UNIVERSITÀ DEGLI STUDI DI SALERNO
DIPARTIMENTO DI MATEMATICA E INFORMATICA

PH. D. THESIS IN MATHEMATICS

SOME TOPICS ON FUZZY LOGIC

Daniele Genito

Ph.D. Director:
P. Longobardi

Advisor:
G. Gerla
2009

CONTENTS

CHAPTER 1
PRELIMINARIES

CHAPTER 2
MODELS AND DEDUCTION APPARATUS FOR FUZZY LOGIC

CHAPTER 3
FUZZY LOGIC PROGRAMMING

CHAPTER 4
KRIPKE-BASED BILATTICE LOGIC

CHAPTER 5
PRESERVATION THEOREMS

Introduction

In this thesis we argue about several aspects of fuzzy logic. More precisely we investigate the following topics:

- the properties preserved by a fuzzy model everytime it is subject to some kind of modification in the framework of the model theory for fuzzy logic proposed in [16],

- fuzzy logic programming, similarity logic and meta-programming to take into account the synonymy relation among predicates in accordance with the ideas proposed by M. S. Ying in [33]

- the connection between fuzzy logic and bilattices theory that represents an interesting tool for the treatment of both truth and grade of information (Ginsberg [20]).

In particular, in chapter 1 we introduce some preliminaries on abstract logic.

In chapter 2, we introduce some basic definitions for a model theory for fuzzy logic as proposed in [16]. In particular, we define the notions of homomorphism, congruence, quotient product, ultraproduct. A basic feature of the proposed approach it's that the valuation structures are not fixed, so they vary in a given type. This gives the basis for the results exposed in chapter 5.

In chapter 3, we introduce some general definitions in fuzzy logic programming, a very promising section of fuzzy logic, whose aim is to build up intelligent data-base systems with "flexible" answers, expert systems able to consider vague predicates and so on, combining the might of logic programming and the big adaptability of fuzzy logic. In particular, we investigate the idea to extend fuzzy logic programming to take into account the synonymy relation among predicates in accordance with the similarity logic proposed by M. S. Ying in [33]. The idea of Ying is that it is possible to relax the application of the inference rules in such a way that it is also admitted an approximate matching of the predicate names. As an example it is admitted that from α and $\alpha' \rightarrow\beta$ we can infer β even in the case that α' is only approximately equal to α. An application to such an idea to logic programming was done in several papers (see [1], [3], [11]) where the definition of synonymy refers to Gödel's norm.

We show (see [13]) that given a fuzzy program in a language $\mathcal{L}$, we can translate it into an equivalent classical program in a suitable (meta-)language $\mathcal{L}_m$. Since the predicate names in $\mathcal{L}$ become constants in $\mathcal{L}_m$, this enables us to admit in $\mathcal{L}_m$ meta-relations (as meta-rules) among predicates. In particular, the meta-relation is the synonymy and this enable us to define a synonymy-sensitive fuzzy logic programming.

We prove that there are at least three reasons in favour of such a logic. The first one is that, differently from the papers [1], [3] and [11], all the triangular norms are admitted. The second is that the resulting notion of fuzzy Herbrand model is uniformly continuous with respect to the synonymy relation (a basic property for a synonymy logic). Finally, another reason is that the resulting logic is a similarity logic in the abstract sense given in [17]. This means that its deduction operator is the closure operator obtained by combining the similarity closure operator with the one-step consequence operator associated with the given fuzzy program.

In our approach we propose simply to add to the meta-language $\mathcal{L}_m$ the predicate symbol *"synonymous"*.

We define a suitable notion of least Herbrand model for the similarity-based logic programming create and we show that we obtain an *abstract synonymy logic programming* and the Herbrand models of such a logic are the fixed points of $T_p \circ SYN$, i.e. the Herbrand models of T_p which are fixed points for *SYN*.

In chapter 4, we investigate about the potentialities of bilattice theory ([20]) for fuzzy logic by proposing and discussing some general definitions. In order to give an example, we apply the resulting apparatus to a Kripke-like logic (see [14]).

In chapter 5 we study the modifications of a fuzzy structure and its properties, and the connections about two of them (via homomorphisms); the idea is to extend to fuzzy logic some preserving theorems of classical first order logic. More in particular, we study the properties preserved by a cut, by quotients or by products and ultraproducts and we investigate about the properties preserved after a "deformation" of a fuzzy model, more precisely, after a modification of the valuation part of such a structure (see [15]).

CHAPTER 1

PRELIMINARIES

1. Bounded lattices and homomorphisms

In this section we will remind some elementary notions in lattice theory.

Definition 1.1. An ordered set $\boldsymbol{L} = (L, \leq)$ is a *lattice* provided that, for any $x, y\in L$, both $Inf(\{x, y\})$ and $Sup(\{x, y\})$ exist. L is *bounded* if there is a greatest element 1 and a least element 0. L is *complete* if $Inf(X)$ and $Sup(X)$ exist for every subset X of L.

It is also useful to represent a lattice as an algebraic structure.

Definition 1.2. $(L, \wedge, \vee, 0, 1)$ is a *bounded lattice* if for every $x, y, z\in L$

(i) $x\vee(y\vee z) = (x\vee y)\vee z$; $x\wedge(y\wedge z) = (x\wedge y)\wedge z$ associativity

(ii) $x\vee y = y\vee x$; $x\wedge y = y\wedge x$ commutativity

(iii) $x\vee x = x$; $x\wedge x = x$ idempotence

(iv) $0\vee x = x$; $1\wedge x = x$.

As it is well known, Definitions 1.1 and 1.2 are in a sense equivalent, in fact the following theorem holds true.

Theorem 1.3. Let the algebraic structure $(L, \wedge, \vee, 0, 1)$ be a bounded lattice and define an ordered set $(L, \leq)$ by putting

$$x \leq y \iff x \wedge y = x .$$

Then the relational structure $(L, \leq, 0, 1)$ is a bounded lattice. Viceversa let the relational structure $(L, \leq, 0, 1)$ be a bounded lattice, then by putting for every $x,y\in L$

$$x\wedge y = Inf(\{x, y\}) \text{ and } x\vee y = Sup(\{x, y\})$$

the resulting algebraic structure (L, $\wedge$, $\vee$, 0, 1) is a bounded lattice such that

$$x \leq y \Leftrightarrow x \wedge y = x.$$

In accordance with such a theorem we can represent a lattice either as ordered structures or as algebraic structures. Neverthless, the two approaches are not equivalent with respect to the notion of homomorphism (and therefore from category point of view).

Definition 1.4. Given two ordered sets L_1 and L_2, a map $h : L_1 \to L_2$ is an *order-homomorphism* if it is order-preserving, i.e., for every $x, y \in L_1$,

$$x \leq y \ \Rightarrow\ h(x) \leq h(y)$$

An *order-isomorphism* is an one-to-one order-homomorphism h whose inverse is an order-homomorphism. An *order-automorphism* of L is an order-isomorphism from L onto itself. We say also that h is an *embedding* if

$$x \leq y \ \Leftrightarrow\ h(x) \leq h(y).$$

Trivially, an embedding is injective and an isomorphism is an one-to-one embedding. The definition of homomorphism in the case the lattices are considered as algebraic structures is the usual one in universal algebra:

Definition 1.5. Given two bounded lattices $\boldsymbol{L}_1 = (L_1, \wedge, \vee, 0, 1)$ and $\boldsymbol{L}_2 = (L_2, \wedge, \vee, 0, 1)$, a map $h : L_1 \to L_2$ is an *algebraic homomorphism* from $\boldsymbol{L}_1$ to $\boldsymbol{L}_2$ if for every x, y in L_1

$$h(0) = 0 \ ;\ h(1) = 1 \ ;\ h(x \wedge y) = h(x) \wedge h(y) \ ;\ h(x \vee y) = h(x) \vee h(y).$$

An *algebraic-isomorphism* is a bijective algebraic-homomorphism and an *algebraic automorphism* in a lattice $\boldsymbol{L}$ is an algebraic-isomorphism from $\boldsymbol{L}$ onto itself.

Theorem 1.6. Let $\boldsymbol{L}_1$ and $\boldsymbol{L}_2$ be two lattices, then every algebraic homomorphism from $\boldsymbol{L}_1$ to $\boldsymbol{L}_2$ is an order-homomorphism from $\boldsymbol{L}_1$ into $\boldsymbol{L}_2$. The viceversa is not valid. The order isomorphisms coincide with the algebraic isomorphisms.

Proof. Let $h : L_1 \to L_2$ an algebraic *homomorphism* from $\boldsymbol{L}_1$ to $\boldsymbol{L}_2$, then

$$x \leq y \Rightarrow x \wedge y = x \ \Rightarrow\ h(x \wedge y) = h(x) \ \Rightarrow\ h(x) \wedge h(y) = h(x) \ \Rightarrow\ h(x) \leq h(y)$$

Consider the lattices ($P\{a,b\}$, $\subseteq$) and ($\{0,1,2\}$, $\leq$), with the map $h : P\{a,b\} \rightarrow \{0, 1, 2\}$such that

$$h(\emptyset) = 0 \ ; \ h(\{a\}) = 1 \ ; \ h(\{b\}) = 1 \ ; \ h(\{a,b\}) = 2$$

Then h is order-preserving but isn't an algebraic homomorphism. In fact

$$h(\{a\}\cup\{b\}) = h(\emptyset) = 0 \neq 1 = 1\wedge 1 = h(\{a\})\wedge h(\{b\}).$$

Assume that $h : L_1 \rightarrow L_2$ is an order-isomorphism and let x and y be elements in L_1. Then, trivially, $h(x\wedge y) \leq h(x)$ and $h(x\wedge y) \leq h(y)$. Assume that $m' \leq h(x)$ and $m' \leq h(y)$ and let m be such that $h(m) = m'$. Then, being h an order isomorphism, $m \leq x$ and $m \leq y$ and therefore $m \leq x\wedge y$. This proves that $m' \leq h(x\wedge y)$ and therefore that $h(x\wedge y)$ is the greatest lower bound of the pair $\{h(x), h(y)\}$. In a similar way one proves the remaining part of the proposition.

We conclude such a section by giving the notion of semilattice.

Definition 1.7. A *bounded semilattice* is an algebraic structure (L, $\wedge$, 0, 1) such that for every $x, y, z \in L$

(i) $x\wedge(y\wedge z) = (x\wedge y)\wedge z$ associativity

(ii) $x\wedge y = y\wedge x$ commutativity

(iii) $x\wedge x = x$ idempotence

(iv) $0\wedge x = 0$; $1 \wedge x = x$.

Given a bounded semilattice (L, $\wedge$, 0, 1) if we define a relation $\leq$ in L by putting $x \leq y \Leftrightarrow x\wedge y = x$, then we obtain a bounded ordered set (L, $\leq$).

2. Closure operators and closure systems

The notions of closure operator and closure system can be defined in any complete lattice L (see [31]). In the following, we call an *operator* in L any map Δ from L into L and *class* or *system* in L any subset C of L.

Definition 2.1 Let L be a complete lattice. Then a *closure operator* in L is any operator $\Delta : L \rightarrow L$ *satisfying* *i)* $x \leq y \Rightarrow \Delta(x) \leq \Delta(y)$ *(order-preserving)*

ii) $x \leq \Delta(x)$ *(inclusion)*

iii) $\Delta(\Delta(x)) = \Delta(x)$ *(idempotence).*

If (iii) is skipped, Δ is called an *almost closure operator* in *L*, in brief *a-c-operator*.

We interpret an element $x \in L$ as a piece of information and $\Delta(x)$ as the whole information we can derive from *x*.

Example. Let $\oplus$ be the set of formulas of a first order logic. Then we can consider the *immediate consequence operator* $\Delta : \Pi(\oplus) \to \Pi(\oplus)$, i.e. the operator defined by setting, for any $X \in \Pi(\oplus)$,

$$\Delta(X) = \{\alpha : \alpha \leftarrow \beta \text{ , } \beta \in X\} \cap \{\forall x \alpha : \alpha \in X\} \cap Al \cap X$$

where *Al* is the set of logical axioms. Then Δ is a *a-c-operator*.

Strictly related with the notion of closure operator, we introduce the one of closure system.

Definition 2.2. A nonempty class *C* of elements of a complete lattice *L* is called a *closure system* if the meet of any class of elements of *C* is an element of *C*.

Observe that, since $Inf(\emptyset) = 1$, every closure system contains 1.

Definition 2.3. Given a closure system *C* and $x \in L$, the *element <x> of C generated by x* is defined by setting

$$<x> = Inf\{x' \in C : x' \geq x\}. \qquad (2.1)$$

The following proposition, whose proof is trivial, shows that any closure system is a complete lattice.

Proposition 2.4. *Let $C \subseteq L$ be a closure system. Then C is a complete lattice such that*

- *the least element of C is the meet of all the elements in C,*
- *the unity in C coincides with the unity* 1 *in L,*
- *the meets in C coincide with the meets in L,*

- the join in C of a class X is $<Sup(X)>$.

Observe that C is not necessarily a sublattice of L and this since the joins in C are different from the joins in L, in general. As an example, consider the class C of subalgebras of a given algebraic structure $\boldsymbol{A}$. Then C is a closure system and hence a complete lattice but, while the meet operator coincides with the intersection, the join of a family $(\boldsymbol{A}_i)_{i\in I}$ of subalgebras coincides with the subalgebra generated by $\cap_{i\in I}\boldsymbol{A}_i$.

The class of closure operators and the class of closure systems define two closure systems in the *direct power* L^L of L with index set L.

Proposition 2.5. *Both the classes $CO(L)$ of closure operators and $AC(L)$ of almost closure operators in L are closure systems in L^L. The class $CS(L)$ of closure systems in L is a closure system in* $\Pi(L)$.

Proof. Let J be the meet of a family $(J_i)_{i\in I}$ of *a-c*-operators and $x \in L$. Then it is immediate that J satisfies (i) and (ii) of Definition 2.1. Assume that each J_i is idempotent. Then, for every $x \in L$ and $k \in I$,

$$J(J(x)) = J(Inf_{i\in I}\, J_i(x)) \leq J_k(Inf_{i\in I}\, J_i(x)) \leq J_k(J_k(x)) = J_k(x).$$

Hence

$$J(J(x)) \leq Inf_{k\in I}\, J_k(x) = J(x)$$

and therefore $J(J(x)) = J(x)$. This proves that $CO(L)$ is a closure system.

Let C be the intersection of a family $(C_i)_{i\in I}$ of closure systems and let $(x_j)_{j\in J}$ be any family of elements in C. We claim that $x = Inf\{x_j : j \in J\}$ is an element of C. Indeed, since, for every $i \in I$, $(x_j)_{j\in J}$ is a family of elements in C_i, we have $x \in C_i$. Thus, $x \in \cup\{C_i : i \in I\} = C$. This proves that $CS(L)$ is a closure system.

3. Connecting the two notions

Let J be an operator in L and C a class of elements of L. Then we denote by $c(J)$ the closure operator generated by J and by $c(C)$ the closure system generated by C.

Now we will show how the closure systems and the closure operators are strictly related. To this purpose we define the map $Co : \Pi(L) \to L^L$ by assuming that, for each $C \subseteq L$, $Co(C) : L \to L$ is the operator defined by setting

$$Co(C)(x) = Inf\{y \in C : y \geq x\}, \qquad (3.1)$$

for every $x \in L$. Moreover, we define the map $Cs : L^L \to \Pi(L)$ by setting, for any operator $J \in L^L$,

$$Cs(J) = \{x \in L : J(y) \leq x \text{ for every } y \leq x\}. \qquad (3.2)$$

Proposition 3.1. *Given any $C \subseteq L$, the operator $Co(C) : L \to L$ is a closure operator. Given any $J \in L^L$, the class $Cs(J)$ is a closure system.*

Proof. The first part is trivial. Let $(x_i)_{i\in I}$ be a family of elements of $Cs(J)$. We claim that $Inf\{x_i : i \in I\} \in Cs(J)$. Indeed, suppose $y \leq Inf\{x_i : i \in I\}$, that is $y \leq x_i$ for every $i \in I$. Then $J(y) \leq x_i$ for every $i \in I$. Thus $J(y) \leq Inf\{x_i : i \in I\}$ and therefore $Inf\{x_i : i \in I\} \in Cs(J)$. This proves that $Cs(J)$ is a closure system.

As claimed in Section 2, sometimes we write $<x>$ instead of $Co(C)(x)$. Given an operator J, we call *fixed point of J* any element $x \in L$ such that $J(x) = x$. In the case that J is an *a-c*-operator this is equivalent to saying that $J(x) \leq x$, i.e., *x is closed with respect to J.* Moreover, we have the following:

Proposition 3.2. *Let J be an a-c-operator. Then*

$$Cs(J) = \{x \in L : J(x) = x\}, \qquad (3.3)$$

i.e., $Cs(J)$ is the class of fixed points of J.

Proof. By (3.2), $J(x) \leq x$ for every $x \in Cs(J)$. Then, if J is an *a-c*-operator, $J(x) = x$. Conversely, if x is a fixed point of J and $y \leq x$, then $J(y) \leq J(x) = x$ and this proves that $x \in Cs(J)$.

The proof of the next proposition is evident:

Proposition 3.3. *Let J and J' be operators and C, C' classes. Then,*

$$J \leq J' \Rightarrow Cs(J) \supseteq Cs(J') \quad ; \quad C \subseteq C' \Rightarrow Co(C) \geq Co(C').$$

The first implication says that, if J and J' are *a-c*-operators such that $J \leq J'$, then every fixed point for J' is a fixed point for J. The following theorem gives a way to obtain the closure operator $c(J)$ generated by J:

Theorem 3.4. *Let J be an operator. Then*

$$c(J) = Co(Cs(J)). \qquad (3.4)$$

So, if J is an a-c-operator and $x \in L$, $c(J)(x)$ is the least fixed point of J greater than or equal to x.

Proof. Set $J' = Co(Cs(J))$, then J' is a closure operator. To prove that $J' \geq J$, it suffices to observe that for every $y \in L$, from $y \in Cs(J)$ and $x \leq y$, it follows that $J(x) \leq y$. Consequently,

$$J'(x) = Inf\{y \in Cs(J) : y \geq x\} \geq J(x).$$

Let H be a closure operator such that $H \geq J$ and suppose $y' \leq H(x)$. Then

$$J(y') \leq H(y') \leq H(H(x)) = H(x).$$

This proves that $H(x) \in Cs(J)$ and, since $H(x) \geq x$, that $H(x) \geq J'(x)$.

Given any class C of elements of L, we can obtain the closure system $c(C)$ generated by C as follows:

$$c(C) = \{Inf(X) : X \subseteq C\}. \qquad (3.5)$$

Moreover, we have the following theorem:

Theorem 3.5. *Given any class C of elements of L, we have*

$$c(C) = Cs(Co(C)). \qquad (3.6)$$

Proof. Being every element of C a fixed point of $Co(C)$, $Cs(Co(C))$ is a closure system containing C. Let C' be a closure system containing C, and x an element of $Cs(Co(C))$. Then, since $x = Co(C)(x)$, x is a meet of elements of C and hence belongs to C. Thus $Cs(Co(C)) \subseteq C'$ and, therefore, $Cs(Co(C)) = c(C)$.

Some interesting properties of the operators Co and Cs are listed in the following proposition:

Proposition 3.6. *Let J and J' be operators, and C and C' classes. Then*

$$Cs(J) = Cs(c(J)) \quad ; \quad Co(C) = Co(c(C)). \qquad (3.7)$$

Also,

$$c(J) = c(J') \Leftrightarrow Cs(J) = Cs(J') \;\; ; \;\; c(C) = c(C') \Leftrightarrow Co(C) = Co(C') \qquad (3.8)$$

Moreover, if C and C' are closure systems, and J and J' closure operators, then

$$C \subseteq C' \Leftrightarrow Co(C) \geq Co(C') \quad ; \quad J \leq J' \Leftrightarrow Cs(J) \supseteq Cs(J') \qquad (3.9)$$

$$Co(Cs(J)) = J \quad ; \quad Cs(Co(C)) = C. \qquad (3.10)$$

4. Abstract logic and continuity

The following is the main definition in this chapter:

Definition 4.1. Let Δ be a closure operator in a complete lattice L. Then we can say that the pair $\Sigma = (L,\Delta)$ is an *abstract deduction system* and that Δ is a *deduction operator* ([4]).

We call *pieces of information* the elements in L. Any classical logic Λ defines an abstract logic whose pieces of information are the set of formulas. Indeed, if $\mathcal{F}$ is the set of formulas of Λ, we can set $L = \Pi(\mathcal{F})$ and Δ equal to the operator associating any $X \in \Pi(\mathcal{F})$ with the set $\Delta(X)$ of consequences of X.

A *theory* in an abstract deduction system (L, Δ) is defined as a fixed point of Δ, i.e., a piece of information τ closed under deductions. Proposition 3.2 says that the class $\mathrm{T} = Cs(\Delta)$ of theories of a deduction system is a closure system and hence a complete lattice. If τ is a theory and $\Delta(x) = \tau$, then we can say that x is a *system of axioms* for τ. A piece of information $x \in L$ is *inconsistent* provided that $\Delta(x) = 1$. This extends the fact that in classical logic an inconsistent set of axioms generates the whole set $\mathcal{F}$ of formulas (i.e., the greatest element of $\Pi(\mathcal{F})$). In accordance, the piece of information 1 is called the *inconsistent theory* and a theory τ is *consistent* provided that $\tau \neq 1$. A *maximal theory* is a theory τ which is maximal in the class of consistent theories, i.e., no theory τ' exists such that $1 > \tau' > \tau$.

Definition 4.2. A class M of elements of L such that $1 \notin \mathrm{M}$ is called an *abstract semantics* and the elements in M are called *models*. If $x \in L$, $m \in \mathrm{M}$ and $x \leq m$, then we can say that m is a *model* of x and we can write $m \leq x$. If $x, y \in L$ admit the same models, then we can say that x is *logically equivalent* to y.

In accordance with Proposition 3.1, any semantics M induces a closure operator $Co(\mathrm{M}) : L \to L$. We call this a *logical consequence operator* and we denote it by Lc. Then, Lc is defined by setting, given a piece of information x,

$$Lc(x) = Inf\{m \in \mathrm{M} : m \leq x\}.$$

These definitions are in accordance with the classical definitions because we can identify the class of models in a classical logic Λ with the class M of complete theories of Λ. In fact, each model m in Λ is associated with its theory, i.e., with the complete theory

$$T_m = \{\alpha \in \mathcal{F} : \alpha \text{ is true in } m\}.$$

Conversely, for every complete theory T a model m exists such that $T_m = T$. Moreover, it is easy to see that m is a model of a set X of formulas iff $X \subseteq T_m$ and that the set $Lc(X)$ of logical consequences of X is equal to the intersection of all the complete theories containing X.

If τ is a theory of Lc we can say also that τ is a theory of M. Trivially,

$$x \text{ is logically equivalent to } y \Leftrightarrow Lc(x) = Lc(y).$$

We can also define Lc as follows: Consider the operators

$$mod : L \rightarrow \Pi(\mathrm{M}) \text{ and } th : \Pi(\mathrm{M}) \rightarrow L$$

defined by setting, for every $x \in L$ and $\mathrm{X} \in \Pi(\mathrm{M})$,

$$mod(x) = \{m \in \mathrm{M} : m \leq x\} \quad ; \quad th(\mathrm{X}) = Sup\{x \in L : m \leq x \ \forall m \in \mathrm{X}\}.$$

Then, $mod(x)$ is the set of models of x and $th(\mathrm{X})$ the information shared by all the models in X. It is easy to verify that mod and th define a Galois connection such that $th \lfloor mod$ coincides with the closure operator Lc.

We define the *system of tautologies* as

$$Tau(\mathrm{M}) = Inf\{m : m \in \mathrm{M}\},$$

equivalently,

$$Tau(\mathrm{M}) = Lc(\emptyset).$$

If x is consistent with respect to Lc, then we prefer to say that x is *satisfiable*. Equivalently, x is satisfiable if a model of x exists. Also, x is *categorical* if just one model of x exists. We denote the class of satisfiable pieces of information by $Sat(\mathrm{M})$, i.e.,

$$Sat(\mathrm{M}) = \{x \in L : m \in \mathrm{M} \text{ exists such that } m \leq x\}.$$

Definition 4.3. An *abstract logic* is a triplet (L, Δ, M) where (L, Δ) is an abstract deduction system and M an abstract semantics such that $\Delta = Co(\mathrm{M})$, i.e., the *"completeness theorem"* holds.

In defining the notion of abstract logic it seems natural to require some additional properties as an example, the basic notion of compactness.

Definition 4.4. Let $J : \Pi(S) \rightarrow \Pi(S)$ be an operator in $\Pi(S)$. Then we say that J is *compact*, provided that, for every subset X of S,

$$x \in J(X) \Leftrightarrow \text{a finite subset } X_f \text{ of } X \text{ exists such that } x \in J(X_f).$$

Equivalently J is compact if

$$J(X) = \cup\{\ J(X_f) : X_f \text{ is finite, } X_f \subseteq X\ \}$$

Due to the finiteness of any proof, the deduction operator of a crisp logic is compact. Now, being the notion of finite subset not defined on a generic lattice L, we cannot define a compactness property in a generic abstract logic. Then we propose the notion of continuity as a natural counterpart of the notion of compactness. To this aim, we give some definitions (see [23]).

Definition 4.5. A nonempty class X of elements in an ordered set L is *upward directed* if

$$x \in \mathrm{X} \text{ and } y \in \mathrm{X} \ \Rightarrow\ \exists\, z \in \mathrm{X}\ ,\ x \leq z \text{ and } y \leq z.$$

If X is upward directed, and $z = Sup(X)$, then we say that z is the *limit* of X and we write $z = limX$.

Obviously the totally ordered subsets of L are examples of upward directed classes. In the sequel, we write "directed" to mean "upward directed".

If J is an order-preserving operator and X is directed, then the image $J(\mathrm{X}) = \{J(x) : x \in \mathrm{X}\}$ is also directed. Then we can give the following definition:

Definition 4.6. An order-preserving operator J is *continuous* if, for every directed class X,

$$J(lim\,\mathrm{X}) = limJ(\mathrm{X}) \qquad (4.1)$$

A continuous closure operator is also called an *algebraic closure* operator.

The following proposition shows that the notion of continuity extends the notion of compactness.

Proposition 4.7. Assume that L is the lattice $\Pi(S)$ of all subsets of a given set S. Then J is continuous iff J is compact.

Proof. If J is continuous and $X \in \Pi(S)$, then, because $C = \{X_f : X_f \text{ is a finite subset of } X\}$ is directed, we have

$$J(X) = J(lim\ C) = lim\ J(C) = \cap\{J(X_f) : X_f \text{ is a finite subset of } X\ \}.$$

Conversely, let $J : \Pi(S) \rightarrow \Pi(S)$ be compact and observe that if C is a directed class of subsets of S then

$$X_f \subseteq lim\ C \text{ and } X_f \text{ finite} \Rightarrow \text{there exists } X \in C \text{ such that } X_f \subseteq X.$$

Consequently,

$$J(lim\ C) = \cap\ \{J(X_f)\ : X_f \subseteq lim\ C\ \}) = \cap\{J(X) : X \in\ C\ \} = lim\ J(C).$$

The connection between closure operators and closure systems suggests the following question: *is there a property for closure systems fitting the continuity property for closure operators well ?*

The next definition enables us to give a positive answer.

Definition 4.8. A class *C* of elements of *L* is called *inductive* if the limit of every directed family of elements in *C* belongs to *C*. An inductive closure system is called *algebraic*.

Every finite subset of *L* is inductive and therefore every finite closure system is algebraic. The notion of an algebraic closure system is well related to the notion of an algebraic closure operator.

Theorem 4.9. Given a nonempty class *C*,

C is an algebraic closure system ⇔ *Co*(*C*) is an algebraic closure operator

Given a closure operator *J*,

$$J \text{ is algebraic } \Leftrightarrow\ Cs(J) \text{ is an algebraic closure system.}$$

Proof. Suppose the *C* is an algebraic closure system and let T be any directed class. Then, since the set H = {*Co*(*C*)(*x*) : *x* ∈ T} is a directed subclass of *C*, *Sup*{*Co*(*C*)(*x*) : *x* ∈ T} is an element of *C* and therefore a fixed point for *Co*(*C*). Then,

$$Co(C)(Sup(\ \mathrm{T})) \subseteq Co(C)(Sup\{Co(C)(x) : x \in\ \ \mathrm{T}\})$$

$$= Sup\{Co(C)(x) : x \in\ \mathrm{T}\},$$

and this proves that *Co*(*C*) is algebraic.

Conversely, let *Co*(*C*) be algebraic and let T be a directed subset of *C*. Then, as *C* is the class of fixed points of *Co*(*C*),

$$Co(C)(Sup(\mathrm{T}))) = Sup(\{Co(C)(x) : x \in\ \mathrm{T}\}) = Sup(\mathrm{T}).$$

This proves that *Sup*(T)) is a fixed point for *Co*(*C*) and hence an element of *C*. In conclusion, *C* is an algebraic closure system.

In order to prove the second part of the proposition, recall that if *J* is a closure operator then *J* = *Co*(*Cs*(*J*)).

Example. Examine the closure systems in the lattice [0,1]. Then a closure system is any subset C of [0,1] closed with respect to the greatest lower bounds. Now, any nonempty subset of [0,1] is directed and therefore the algebraic closure systems coincide with the subsets C both closed under least upper bounds of subsets and greatest lower bounds of nonempty subsets. Thus, the class of algebraic closure systems coincides with $\{X \subseteq [0,1] : X$ is closed and $1 \in X\}$. For instance, set $C = [1/3, 2/3) \cup \{1\}$. Then C is a closure system which is not algebraic and the associated closure operator is defined by setting:

$Co(C)(x) = 1/3$ for every $x \in [0,1/3]$,

$Co(C)(x) = x$ for every $x \in [1/3, 2/3)$,

$Co(C)(x) = 1$ otherwise.

Moreover, we have that its topological closure $[1/3, 2/3] \cup \{1\}$ is an algebraic closure system. Notice that the continuity proposed in Definition 4.6 is different than the continuity with respect to the natural topology in [0,1]. In fact, an operator J satisfies (4.1) iff J is order-preserving and lower semicontinuous with respect to natural topology.

We conclude this section with the following basic definition:

Definition 4.10. An abstract deduction system (L, Δ) (more generally, an abstract logic) is called *continuous* provided that Δ is continuous.

5. Fixed points and step-by-step deduction systems

Usually a deduction operator Δ is defined by giving a suitable set A of logical axioms and a suitable set of inference rules. In this case we can define the *immediate consequence operator H* by setting, for any set X of formulas, $J(X)$ equal to the set of formulas that can be obtained by one application of the inference rules to formulas in X and

$$H(X) = J(X) \cup \mathrm{A} \cup X.$$

In other words, $\alpha \in H(X)$ if either α is obtained by applying an inference rule to formulas in X, or α is a logical axiom or α is a hypothesis (a proper axiom). Also, we define H^n by induction on n, by setting

$$H^1 = H \text{ and } H^{n+1} = H \lfloor H^n.$$

Given a natural number n, $H^n(X)$ represents the set of formulas that can be achieved by an n-step inferential process from X. It is easy to prove that H is a compact almost closure operator and that Δ is the closure operator generated by H. Moreover,

$$\Delta(X) = \cap_{n \in \boldsymbol{N}} H^n(X).$$

To extend such an approach to abstract logics, we must first examine how to obtain the closure operator generated by a continuous *a*-*c*-operator.

Proposition 5.1. Let H be a continuous a-c-operator. Then the set $Cs(H)$ of fixed points of H is an algebraic closure system and the closure operator $c(H)$ generated by H, is an algebraic closure operator.

Proof. Let T be a directed subclass of $Cs(H)$. Then, from the continuity of H it follows that:

$$H(Sup(\mathrm{T})) = Sup(\{H(x) : x \in \mathrm{T}\}) = Sup(\mathrm{T})$$

and, hence, $Sup(\mathrm{T}) \in Cs(H)$. This proves that $Cs(H)$ is algebraic. Thus, from the equality, $c(H) = Co(Cs(H))$, we can conclude that $c(H)$ is algebraic.

Let H be a continuous *a*-*c*-operator. Then, the following simple and useful theorem enables us to calculate the closure operator $c(H)$ generated by H (see, for example, [23]).

Theorem 5.2. (*Fixed-Point Theorem*). Let H be a continuous a-c-operator. Then

$$c(H) = Sup_{n\in N} H^n. \qquad (5.1)$$

In other words, for every $x \in L$, the least fixed point of H greater than or equal to x is given by $Sup_{n\in N} H^n(x)$.

Proof. We have to prove that, for every $x \in L$, $Sup_{n\in N}H^n(x)$ is the least fixed point of H greater than or equal to x. Now, the inequality $H(x) \geq x$ entails that $H^{n+1}(x) \geq H^n(x)$ for every n, and hence, that $(H^n(x))_{n\in N}$ is directed. By the continuity of H,

$$H(Sup_{n\in N}H^n(x)) = Sup_{n\in N}H^{n+1}(x) = Sup_{n\in N}H^n(x)$$

and $Sup_{n\in N}H^n(x)$ is a fixed point for H greater than or equal to x. Let y be any fixed point such that $y \geq x$. Then, for every $n \in N$, $y = H^n(y) \geq H^n(x)$ and therefore,

$y \geq Sup_{n\in N}H^n(x)$. This proves that $Sup_{n\in N} H^n(x) = c(H)(x)$.

In accordance with the above considerations, we propose the following definition extending the example in Section 2:

Definition 5.3. An *abstract step-by-step* deduction system is an triplet like (L, J, a) where

- L is a complete lattice,
- J is a continuous operator in L,
- a is an element of L (the system of *logical axioms*).

Let (L, J, a) be a step-by-step deduction system and define H by setting

$$H(x) = J(x) \vee x \vee a, \qquad (5.2)$$

for every $x \in L$. Then, H is a continuous *a-c*-operator we call *the immediate consequence operator.*

Definition 5.4. Let (L, J, a) be a step-by-step-deduction system and denote by Δ the closure operator generated by the immediate consequence operator H. Then the abstract deduction system (L, Δ) is called the *deduction system associated with* (L, J, a).

The proof of the following theorem is trivial:

Theorem 5.5. *Let* (L, J, a) *be a step-by-step-deduction system and* (L, Δ) *the associated deduction system. Then* Δ *is continuous and*

$$\Delta(x) = Sup_{n\in N}H^n(x). \qquad (5.3)$$

Moreover, τ *is a theory of* (L, Δ) *iff* $\tau \geq J(\tau)$ *and* $\tau \geq a$.

6. The product of two deduction systems

Given two abstract deduction systems (L,Δ) and (L,Δ'), it is natural to search for a new deduction apparatus able to use both the inferential instruments of (L,Δ) and (L,Δ'). This suggests considering the operators $\Delta \lfloor \Delta'$, $\Delta \lfloor \Delta'$ and $\Delta \vee \Delta'$. Now, the composition (and the join) of two closure operators is, in general, an almost closure operator and not a closure operator. Consequently, we have to refer to the closure operators generated by these operators which coincide as the following theorem shows:

Theorem 6.1. *Let* J *and* J' *be a-c-operators. Then*

$$Cs(J \lfloor J') = Cs(J) \cap Cs(J') = Cs(J \vee J') = Cs(J' \lfloor J). \qquad (6.1)$$

Consequently,

$$c(J \lfloor J') = c(J' \lfloor J) = c(J \vee J'), \qquad (6.2)$$

i.e., $J \lfloor J'$, $J' \lfloor J$ *and* $J \vee J'$ *generate the same closure operator.*

Proof. Let x be a fixed point of $J \lfloor J'$. Then $J'(x) \leq J(J'(x)) = x$ and therefore x is a fixed point of J'. Moreover, the equalities $x = J(J'(x)) = J(x)$ show that x is also a fixed point of J. Conversely, it is apparent that if x is fixed for both J and J', then x is fixed for $J \lfloor J'$ and this proves the first equality. The remaining part of the proposition follows from (6.1) and Theorem 3.4.

Definition 6.2 Let (L, Δ) and (L, Δ') be two deduction systems. Then we call *product* of (L, Δ) and (L, Δ') the deduction system

$$(L, c(\Delta \lfloor \Delta'\,)) = (L, c(\Delta' \lfloor \Delta)) = (L, c(\Delta \vee \Delta'\,))\ .$$

From the first equality in (6.1) it follows that a piece of information x is a theory of the product $(L, c(\Delta \lfloor \Delta'))$ iff x is a theory of both the deduction system (L, Δ) and (L, Δ'). The proof of the following theorem is evident:

Theorem 6.3. The product of deduction systems is a commutative and associative operation. Moreover, the product of two continuous (logically compact) deduction systems is a continuous (logically compact) deduction system.

Sometimes it is possible that the composition of two closure operators is a closure operator. The following theorem gives some information to this regard:

Theorem 6.4. Let J and J' be closure operators. Then the following are equivalent:

(i) $J \lfloor J'$ is a closure operator.

(ii) $J \lfloor J' \geq J' \lfloor J$.

(iii) $J(J'(x))$ is a fixed point of J' for every $x \in L$.

Proof. (i) ⇒ (ii). Let $x \in L$. Then, from $J'(x) \geq x$ it follows that $J(J'(x)) \geq J(x)$ and therefore $J'(J(J'(x))) \geq J'(J(x))$. Thus, since by hypothesis $J \lfloor J'$ is a closure operator, by the inclusion property for J,

$$J(J'(x)) = J(J'(J(J'(x)))) \geq J'(J(J'(x))) \geq J'(J(x)).$$

(ii) ⇒ (iii). Observe that

$$J'(J(J'(x))) \leq J(J'(J'(x))) = J(J'(x)).$$

(iii) ⇒ (i). Observe that

$J(J'(J(J'(x)))) = J(J(J'(x))) = J(J'(x))$

7. Triangular norms and co-norms

In fuzzy sets theory, triangular norms are usually used to generalize the logical conjunction “and”. More precisely, we have the following definition:

Definition 7.1. A map T: $[0,1] \times [0,1] \rightarrow [0,1]$ is called *triangular norm* (*or T-norm*) if satisfies the following properties:

i) $T(a,b) = T(b,a)$ (commutativity)

ii) $T(a, T(b, c)) = T(T(a, b), c)$ (associativity)

iii) $T(a, b) \leq T(c, d)$ if $a \leq c$ and $b \leq d$ (monotonicity)

iv) $T(a,1) = a$ (identity element)

$\forall\, a,b,c,d \in [0,1]$.

The most used t-norm are:

- minimum (or Gödel): $min(a, b) = min\{a, b\}$
- Łukasiewicz: $T_L(a, b) = max\{a + b - 1, 0\}$
- product: $T_P\,(a, b) = ab$

If T is a t-norm, and h: $[0,1] \rightarrow [0,1]$ is an increasing bijection, then

$$T^*(a,b) = h^{-1}(T(h(a),h(b)))$$

is a t-norm.

The dual concept is the notion of triangular co-norms that instead are extensively used to model logical connectives “or”.

Definition 7.2. A map $S : [0,1] \times [0,1] \rightarrow [0,1]$ is a *triangular co-norm (t-conorm)* if it is symmetric, associative, nondecreasing in each argument and $S(a, 0) = a$, for all $a \in [0, 1]$. In other words, any t-conorm S satisfies the properties:

i) $S(a,b) = S(b,a)$ (commutativity)

ii) $S(a, S(b, c)) = S(S(a, b), c)$ (associativity)

iii) $S(a, b) \leq S(c, d)$ if $a \leq c$ and $b \leq d$ (monotonicity)

iv) $S(a, 0) = a$ (zero identity)

$\forall a,b,c,d \in [0, 1]$

If T is a t-norm, it’s possible to define a t-conorm S associated to T by the equality

$$S(a, b) := 1- T(1 - a, 1 - b)$$

and we say that S is derived from T. The basic t-conorms are:

- maximum: $max(a, b) = max\{a, b\}$
- Łukasiewicz: $S_L(a, b) = min\{a + b, 1\}$
- probabilistic: $S_P\,(a, b) = a + b - ab$

Definition 7.3. A t-norm is called *continuous* if it is continuous as a function, in the usual interval topology on $[0, 1]^2$.

For any left-continuous t-norm T, there is a unique binary operation I: $[0,1] \to [0,1]$ such that

$T(x,z) \leq y$ if and only if $z \leq I\,(x,y)\ \ \forall\, x, y, z \in [0,1]$

This operation is called the *residuum* of the t-norm and is frequently denoted by $\to$ since in a t-norm based fuzzy logics, if the logic conjunction is interpreted by a t-norm, the implication is interpreted by the residuum. Moreover, observe that the interval [0, 1] equipped with a t-norm and its residuum is a residuated lattice.

More in general we have the following definition:

Definition 7.4. We call *residuated lattice* the structure $\boldsymbol{L} = (L, \wedge, \vee, \otimes, \to, 0, 1)$ where $(L, \wedge, \vee, 0, 1)$ is a bounded lattice, $\otimes$ is a commutative, associative, order-preserving binary operation whose neutral element is 1, $\to$ is a binary operation such that

$$x \otimes z \leq y \iff z \leq x \to y.$$

Observe that if $\otimes$ is sup-preserving and $(L, \wedge, \vee, 0, 1)$ is complete, then we obtain a residuated lattice by defining $\to$ by the equation

$$x \to y = Sup\{z \in L : x \otimes z \leq y\}.$$

If we refer to the class of the structures in which $\to$ is defined in such a way, then a function f from a residuated lattice $\boldsymbol{L} = (L, \wedge, \vee, \otimes, \to, 0, 1)$ into a residuated lattice $\boldsymbol{L'} = (L', \wedge, \vee, \otimes', \to, 0, 1)$ is an isomorphism if an only if f is an isomorphism from to the reduct $(L, \wedge, \vee, \otimes, 0, 1)$ to the reduct $(L', \wedge, \vee, \otimes', 0, 1)$.

The operations $\otimes$ and $\to$ are used to interpret a *conjunction* $\underline{\wedge}$ and an *implication* $\underline{\to}$, respectively. In a residuated lattice we can define an operation ~ by setting $\sim(\lambda) = \lambda \to 0$. Such an operation is the interpretation of the negation $\neg$.

8. Similarity

Definition 8.1. Let S be a nonempty set and $\otimes$ a triangular norm, then a fuzzy relation $R : S{\times}S{\to}[0,1]$ is a *similarity* if for every x, y, z in S,

(a) $R(x,x) = 1$ (reflexivity),

(b) $R(x,y) = R(y,x)$ (symmetry),

(c) $R(x,y) \geq R(x,z) \otimes R(z,y)$ (transitivity).

So a similarity relation can be link in a sense the “similar” elements and can be seen as a weakening of the identity relation. Since the notion of similarity depends strongly from the operation $\otimes$, to emphasize such a dependence sometime we say that R is a $\otimes$-similarity.

In logic programming a similarity relation can be used to modify the classical unification in a “relaxed unification” which is particulary interesting when, in the classical unification process, a failure happens.

Definition 8.2. For every $\lambda \in [0,1]$, the *λ-cut* of a similarity relation R is the set $R_\lambda = \{(x, y) \in S{\times}S \mid R(x, y) \geq \lambda \}$.

If $(x, y) \in R_\lambda$ we can say that x is λ-similar to y.

CHAPTER 2

MODELS AND DEDUCTION APPARATUS FOR FUZZY LOGIC

1. Fuzzy interpretations of a first order language

In this chapter we recall some basic notions in fuzzy logic. In particular, we define several model theoretic notions in accordance with the approach proposed in [16].

In order to evaluate the formulas in a multi-valued logic, we need a set V of truth values and suitable operations in V able to interpret the logical connectives. An order relation in V enables us to interpret the universal and existential quantifiers by the least upper bound and greatest lower bound operators, respectively. Technical reasons suggest to introduce such an order by a semilattice operation. The following is a more precise definition.

Definition 1.1. A *type for a valuation structure* is a pair $\tau = (\underline{C}, ar)$ defined by a nonempty set $\underline{C}$ and an *arity* function $ar : \underline{C} \to N_0$. If $\underline{\lambda}$ is an element in $\underline{C}$ such that $ar(\underline{\lambda}) = 0$, then $\underline{\lambda}$ is named a *logical constant*. If $\underline{c}$ is an element in C such that $ar(\underline{c}) = n \neq 0$, then $\underline{c}$ is called an *n-ary logical connective*. In the case $ar(\underline{c}) = 2$ we say that $\underline{c}$ is *binary*. We assume that there are at least two logical constants $\underline{0}$ and $\underline{1}$ and a binary logical connective $\underline{\wedge}$ we call *conjunction*.

Definition 1.2. A *valuation structure* of type τ is a pair $\boldsymbol{V} = (V, I)$, where V is a nonempty set (the *true values set*) and I (the *interpretation*) is a map in $\underline{C}$ such that:

a) for every logical constant $\underline{\lambda}$, $I(\underline{\lambda})$ is an element in V,

b) for every n-ary logical connective $\underline{o}$, $I(\underline{o})$ is an n-ary operation in V,

c) if $\wedge = I(\underline{\wedge})$, $0 = I(\underline{0})$ and $1 = I(\underline{1})$, then $(V, \wedge, 0, 1)$ is a semilattice

Then, a valuation structure is an algebraic structure admitting as a reduct a bounded semilattice. As it is usual, we can represent a valuation structure $\boldsymbol{V} = (V, I)$ by the associate algebraic structure. In the case $\underline{C}$ is finite, we write $(V, h_1,\ldots,h_t)$ to denote such a structure. Let $\leq$ denote the order relation defined by setting $x \leq y$ if and only if $x \wedge y = x$. We call *complete* a valuation structure which is complete with respect to such an order. Notice that we admit also incomplete valuation structures since there is a large class of fuzzy theories in which this does not create difficulties. Another reason in favor of such a choice is that otherwise should be impossible to give the notion of quotient. In fact there are complete semilattices that admit quotients aren't complete. Obviously, as we'll see afterward, the incompleteness determines some difficulties in evaluating the quantifiers.

Definition 1.3. A *first order language* L for a fuzzy logic is a system ($\underline{F}$, $\underline{R}$, $\underline{C}$, *ar*) where $\underline{F}$, $\underline{R}$, $\underline{C}$ are disjoint sets and $ar : \underline{F}\cup\underline{R}\cup\underline{C} \to N_0$ is a function we call *arity function* in such a way that ($\underline{C}$, *ar*) is a type for a valuation structure. If $\underline{c} \in \underline{F}$ and $ar(\underline{c}) = 0$, then $\underline{c}$ is called a *constant*. If $\underline{h} \in \underline{F}$ is such that $ar(\underline{h}) = n \neq 0$, then $\underline{h}$ is called an n-ary *operation symbol*. If $\underline{r} \in \underline{R}$ and $ar(\underline{r}) = n$, then $\underline{r}$ will be called an n-ary *predicate symbol* (we assume that the arity of a predicate symbol is different from 0).

Then a first order language is a first order language as usually defined in classical logic together with a type for valuation structures. The semantics for first order multi-valued logic is based on the notion of fuzzy set and fuzzy relation [34].

Definition 1.4. Given a valuation structure $\boldsymbol{V}$ and a nonempty set S, we call *V–subset* or simply *fuzzy subset of* S any map $s : S \to V$ from S to V. For every $x\in S$, the value $s(x)$ is interpreted as a membership degree. An *n-ary V-relation* in S is a V-subset of S^n, i.e. a map $s : S^n \to V$.

The *support* of s is the set $supp(s) = \{x\in S : s(x) \neq 0\}$. A fuzzy subset s is called *crisp* provided that $s(x)\in\{0,1\}$ for every $x\in S$. We say that s is finite provided that its support is finite. We denote by V^S the class of all the fuzzy subsets of S and we identify the subsets of S with the crisp fuzzy subsets by associating every subset with the related characteristic function. In the case $\boldsymbol{V}$ is complete**,** if $(s_i)_{i\in I}$ is a family of fuzzy subsets of S, then $\cup_{i\in I}s_i$ and $\cap_{i\in I}s_i$ are the fuzzy subsets defined by the equations

$$(\cup_{i\in I} s_i)(x) = Sup_{i\in I}\, s_i(x)\ ;\ (\cap_{i\in I} s_i)(x) = Inf_{i\in I}\, s_i(x).$$

Definition 1.5. Given a first order language L, a *fuzzy interpretation of* L is a triple $M = (D, V, I)$ such that D and V are nonempty sets (the *domain* and the *truth values set*, respectively), I (the *interpretation*) is a map such that:

i) V together the restriction of I to $\underline{C}$ is a valuation structure**,**

ii) I associates every n-ary operation symbol $\underline{h}\in\underline{F}$ with an n-ary operation $h =$
$I(\underline{h})$ in D,

iii) I associates every n-ary predicate symbol $\underline{r}\in\underline{R}$ with an n-ary V-relation $r =$
$I(\underline{r})$ in D.

Then a fuzzy interpretation M is defined by assigning:

- a classical algebraic structure ***Al***(M) = (D,F) we call *algebraic structure of the domain*
- a valuation structure ***VAL*(**M) = (V,C)
- a set ***Rel***(M) of fuzzy relations.

A fuzzy interpretation is also called a *first order fuzzy structure*. We say that two fuzzy structures are of the *same type* if they are fuzzy interpretations of the same language. We call *crisp* a fuzzy structure M such that all the fuzzy relation in ***Rel***(M) are crisp.

Definition 1.6. Assume that in the language there is the special relation symbol "=". Then we call *normal* a fuzzy interpretation such that $I(=)$ is the (characteristic function of the) identity relation.

It is evident that we can identify the usual structures in classical logic with the normal crisp fuzzy structures.

Given a first order language $\mathcal{L}$ in which we assume the universal quantifier $\forall$ as a primitive, we indicate with $Form(\mathcal{L})$ the set of all formulas and with $Form(\mathcal{L}_n)$ (with Ter_n) the set of formulas (terms) whose free variables are in $\{x_1,...,x_n\}$. Given a fuzzy structure $M = (D, V, I)$, the interpretation of a term $t \in Ter_n$ is an n-ary function $I(t)$ in D defined by recursion on the complexity of t as in classical logic. The valuation of the formulas of $\mathcal{L}$ with respect to $M = (D, V, I)$ is defined in a truth-functional way as follows.

Definition 1.7. Given a fuzzy structure $M = (D, V, I)$ and $\alpha \in Form(\mathcal{L}_n)$, the *value of* α *in* $d_1,\dots,d_n$ *with respect to* M is the element $val(M, \alpha, d_1,\dots,d_n)$ in V defined, by recursion on the complexity of α, by the equations:

(i) $val(M, \underline{r}(t_1,\dots,t_p), d_1,\dots,d_n) = I(\underline{r})(I(t_1)(d_1,\dots,d_n),...,I(t_p)(d_1,...,d_n))$

(ii) $val(M, \underline{c}(\alpha_1,\dots,\alpha_q), d_1,\dots,d_n) = I(\underline{c})(val(M,\alpha_1,d_1,...,d_n),\dots,val(M,\alpha_q,d_1,\dots,d_n))$

(iii) $val(M, \forall x_h\beta, d_1,\dots,d_n) = Inf(\{val(M, \beta, d_1,\dots,d_{h-1}, d, d_{h+1},\dots,d_n) : d \in D\})$

where $p, q \in N \setminus \{0\}$, $\underline{r} \in \underline{R}_p$, $\underline{c} \in \underline{C}_s$, $t_1,\dots, t_p \in Ter_n$, $\alpha_1,\dots,\alpha_q$, $\beta \in Form(\mathcal{L}_n)$, $h \in \{1,\dots,n\}$.

It is evident that if α is a closed formula, then the value $val(M, \alpha, d_1,\dots, d_n)$ does not depend on the elements $d_1,..,d_n$. In such a case we write $val(M, \alpha)$ instead of $val(M, \alpha, d_1,\dots,d_n)$. In the case $\forall x_1\dots\forall x_n(\alpha))$ is the universal closure of α, we write $val(M, \alpha)$ to denote $val(M, \forall x_1\dots\forall x_n(\alpha))$. Observe that in the case the valuation structure is not complete, due to the presence of the operator *Inf* it is possible that the valuation is undefined for some universal formula.

Definition 1.8. If in a fuzzy structure M all the formulas have a valuation, then M is called a *safe structure.*

Trivially, if $Inf(X)$ exists for every X in $P(V)$, then M is safe. Nevertheless we can refer only to a particular class of subsets of V.

Definition 1.9. Let α be a formula such that $val(M, \alpha, d_1,\dots,d_m)$ exists for every $d_1\in D,\dots,d_m \in D$. Then we call *range of* α *in* M the subset $V(\alpha)$ of V defined by

$$V(\alpha) = \{val(M, \alpha, d_1,\dots,d_m) : d_1\in D,\dots,d_m \in D\}.$$

We denote by $P_M(V)$ the class of all the ranges of the formulas in $M.$

Observe that $P_M(V)$ is enumerable and therefore that, in the case V infinite, $P_M(V)$ is different from the power set $P(V)$ of V. Obviously, it is sufficient to require that $Inf(X)$ exists for every X in $P_M(V)$, to obtain the safeness of $M.$

Definition 1.10. We call *fuzzy theory* any fuzzy subset τ of formulas. We say that a safe interpretation M is a *fuzzy model of* τ, in brief $M \vDash \tau$, if $val(M, \alpha) \geq \tau(\alpha)$ for every formula $\alpha.$

An equivalent formulation of the notion of fuzzy model of τ is obtained by the notion of fuzzy formula.

Definition 1.11. We call *fuzzy formula* a pair $<\alpha,\lambda>$ where α is a formula and $\lambda\in V$. We say that such $<\alpha,\lambda>$ *is satisfied by* M, in brief $M \vDash <\alpha,\lambda>$, if $val(M, \alpha)$ is defined and $val(M, \alpha) \geq \lambda$. We identify the fuzzy formula $<\alpha,1>$ with the formula α and we write $M \vDash \alpha$ if $val(M, \alpha) = 1$.

Then we can represent a fuzzy theory τ as the set

$\{<\alpha, \lambda> : \alpha$ is a formula and $\lambda = \tau(\alpha) \neq 0\}$

of fuzzy formulas and we can say that M is a model of τ if M is a model of all the fuzzy formulas in τ. In the case the support of τ is finite, we can represent τ by a list as

α_1 $[\lambda_1]$

. . .

α_n $[\lambda_n]$

In the case τ is a crisp fuzzy subset, then we can represent τ by the related support $T = \{\alpha : \tau(\alpha) = 1\}$ and we have that (D,V,I) is a model of τ provided that $val(M, \alpha) = 1$ for every $\alpha \in T$. If T is an universal theory, this is equivalent to say that, given any formula $\forall x_1 ... \forall x_n \alpha$ in T,

$val(M, \alpha, d_1,\dots,d_n) = 1$ for every $d_1,\dots,d_n$ in D.

Consequently, in such a case the completeness of the valuation structure plays no role.

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 $\tau(\varphi)$"*.

In accordance with the definition given for an abstract logic, it is possible to define a this a *logical consequence operator* that we denote by *Lc*.

Definition 1.12. 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:

$$Lc(\tau)(\varphi) = Inf\{m(\varphi) : m \vDash \tau\}.$$

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.

2. Homomorphisms and quotients

In the class of fuzzy structures of the same type we can define the notion of homomorphism. To simplify our notation, given a map $h : D \to D'$, we will denote again by h the map $h: D^n \to D'^n$ defined by setting $h(d_1,\dots,d_n) = (h(d_1),\dots,h(d_n))$ for every $(d_1,..,d_n) \in D^n$.

Definition 2.1. Let $M = (D, V, I)$ and $M' = (D', V', I')$ be two fuzzy structures of the same type. Then we say that a pair (h, k) is a *weak homomorphism* from M to M' provided that h is a homomorphism from $\boldsymbol{Al}(M)$ into $\boldsymbol{Al}(M')$, k is a homomorphism from $\boldsymbol{VAL}(M)$ into $\boldsymbol{VAL}(M')$ and, for every predicate symbol $\underline{r}$,

$$k \circ I(\underline{r}) \leq I'(\underline{r}) \circ h.$$

We say that a weak homomorphism (h, k) is a *homomorphism* if

$$k \circ I(\underline{r}) = I'(\underline{r}) \circ h$$

for every predicate symbol $\underline{r}$ (different from the special symbol = in the case M is normal).

As usual, we can express the condition $k \circ I(\underline{r}) \leq I'(\underline{r}) \circ h$ and the identity $k \circ I(\underline{r}) = I'(\underline{r}) \circ h$ by saying that the diagram

$$\begin{tikzcd}
D^n \arrow[r, "h"] \arrow[d, "I(\underline{r})"'] & D'^n \arrow[d, "I'(\underline{r})"] \\
V \arrow[r, "k"] & V'
\end{tikzcd}$$

quasi commutes or *commutes* respectively.

Due to the presence of the semilattice operation $\wedge$, and to the meaning of the constants $\underline{0}$ and $\underline{1}$, if (h,k) is a weak homomorphism, then k is order-preserving and $k(0) = 0$, $k(1) = 1$. In the case k injective,

$$x \leq y \Leftrightarrow k(x) \leq k(y).$$

If there is no algebraic structure in the considered fuzzy structures, then the condition that h is a homomorphism from ***Al***(M) into ***Al***(M') is skipped and the only request for h is the commutativity of the diagram for every fuzzy relation.

Definition 2.3. Let (h,k) be a homomorphism, then

- (h,k) is an *isomorphism* if both h and k are isomorphisms
- (h,k) is an *epimorphism* if both h and k are epimorphisms

In the sequel, if X is a set we indicate by i_X the identity map in X.

Definition 2.4. Let (h,k) be a (weak) homomorphism, then

- (h,k) is a (*weak*) *structure-homomorphism* if $V = V'$ and $k = i_V$
- (h,k) is a (*weak*) *valuation-homomorphism* if $D = D'$ and $h = i_D$

We denote by h the structure homomorphism (h,i_V) and by k the valuation homomorphism (i_D, k).

The second basic notion we have to define is the one of congruence and the related notion of quotient of a fuzzy structure.

Definition 2.5. A *congruence* $\equiv$ in a fuzzy structure $\mathcal{M} = (D,V,I)$ is a pair $(\equiv_1, \equiv_2)$ of congruences of $\boldsymbol{Al}(\mathcal{M})$ and $\boldsymbol{VAL}(\mathcal{M})$, respectively, such that for every m-ary relation symbol $\underline{r}\in\underline{R}$ (different from the special symbol = in the case $\mathcal{M}$ is normal) and for every $d_1,\ldots,d_m, b_1,\ldots,b_m \in D$

$$d_1 \equiv_1 b_1,\ldots, d_m \equiv_1 b_m \ \Rightarrow\ I(\underline{r})(d_1,\ldots,d_m) \equiv_2 I(\underline{r})(b_1,\ldots,b_m) \qquad (2.1)$$

We say that $(\equiv_1, \equiv_2)$ is a *structure congruence* if $\equiv_2$ is the identity relation, we say that $(\equiv_1 , \equiv_2)$ is a *valuation congruence* if $\equiv_1$ is the identity relation.

The condition $\underline{r}$ different from = is a necessary one since otherwise the only possible structure congruence in a normal fuzzy structure is the identity. Indeed

$$b \equiv d \ \Rightarrow\ b \equiv b,\ b \equiv d \Rightarrow 1= I(=)(b,b) = I(=)(b, d) \Rightarrow b = d.$$

Obviously, the valuation congruences coincide with the congruences in $\boldsymbol{VAL}(\mathcal{M})$. Indeed in such a case (2.1) is trivial.

It is useful to consider the class of congruences in a fuzzy structure by referring to the complete Boolean algebra $(Rel(\mathcal{M}), \leq)$ where

$$Rel(\mathcal{M}) = \{(R_1, R_2) : R_1 \in P(D\times D) \text{ and } R_2 \in P(V\times V)\}$$

and where $\leq$ is defined by setting

$$(R_1,R_2) \leq (R'_1, R'_2) \ \Leftrightarrow\ R_1 \subseteq R'_1 \text{ and } R_2 \subseteq R'_2.$$

In other words, such a Boolean algebra is the product of the Boolean algebra of the binary relations in D and the Boolean algebra of the binary relations in V. It is immediate that every congruence is the join in $Rel(\mathcal{M})$ of a structure congruence with a valuation congruence.

Proposition 2.6. The class of congruences of $\mathcal{M}$ is a closure system in the Boolean algebra $(Rel(\mathcal{M}), \leq)$.

Proof. The maximum $(D\times D, V\times V)$ is a congruence and therefore the meet of the empty class is a congruence. Consider a family $(\equiv_1^i, \equiv_1^i)_{i\in I}$ of congruences and consider the related meet $(\equiv_1, \equiv_2)$ in $(Rel(\mathcal{M}), \leq)$

$$(\equiv_1, \equiv_2) = \wedge_{i\in I}(\equiv_1^i,\equiv_2^i) = (\cap_{i\in I} \equiv_1^i, \cap_{i\in I}\equiv_2^i).$$

It is immediate that $\equiv_1$ is a congruences of $\boldsymbol{Al}(\mathcal{M})$ and the $\equiv_2$ is a congruence of $\boldsymbol{V}(\mathcal{M})$. Moreover, for every m-ary relation symbol $\underline{r}$ and $d_1,\ldots,d_m, b_1,\ldots,b_m \in D$,

$$d_1 \equiv_1 b_1,\ldots, d_m \equiv_1 b_m \Rightarrow d_1 \equiv_1^i b_1,\ldots., d_m \equiv_1^i b_m \quad \text{for every } i\in I$$

$$\Rightarrow \quad I(\underline{r})(d_1,\ldots,d_m) \equiv_2^i I(\underline{r})(b_1,\ldots,b_m) \ \text{ for every } i\in I$$

$$\Rightarrow \quad I(\underline{r})(d_1,\ldots,d_m) \equiv_2 I(\underline{r})(b_1,\ldots,b_m).$$

The following proposition gives a way to obtain the congruence generated by a given pair (R_1, R_2).

Proposition 2.7. Given (R_1,R_2) in *Rel*(M), we can obtain the congruence $(\equiv_1, \equiv_2)$ generated by (R_1,R_2) by setting $\equiv_1$ equal to the congruence in ***Al***(M) generated by R_1 and $\equiv_2$ equal to the congruence in ***VAL***(M) generated by

$R_2\cap\{(I(\underline{r})(d_1,\ldots,d_m), I(\underline{r})(b_1,\ldots,b_m)) : \underline{r}\in\underline{R}$ and $d_1\equiv_1 b_1 ,\ldots,d_m\equiv_1 b_m \}$.

Proof. Observe that, by definition, $(\equiv_1,\equiv_2)$ is a congruence containing (R_1,R_2). Let $(\equiv'_1,\equiv'_2)$ be any congruence containing (R_1,R_2). Then $\equiv_1\subseteq\equiv'_1$ because $\equiv_1$ is the smallest congruence of of ***Al***(M) containing R_1 . On the other hand, since $\equiv_1 \subseteq \equiv'_1$ and $(\equiv'_1,\equiv'_2)$ is a congruence, if $d_1\equiv_1 b_1 ,\ldots,d_m\equiv_1 b_m$, then $(I(\underline{r})(d_1,\ldots,d_m) \equiv'_2 I(\underline{r})(b_1,\ldots,b_m))$. Since by hypothesis R_2 is contained in $\equiv'_2$, this proves that $\equiv_2 \subseteq\equiv'_2$.

To define the quotient of a fuzzy structure modulo a congruence, if x and λ are elements in D and V, then we denote by $[x]$ and $[\lambda]$ the equivalence classes modulo $\equiv_1$ and $\equiv_2$, respectively.

Definition 2.8. Let $M = (D,V,I)$ be a fuzzy structure and $\equiv$ a congruence in M. Then the *quotient of M modulo* $\equiv$ is the fuzzy structure $M/\equiv$ such that ***Al***($M/\equiv$) is the quotient of ***Al***(M) modulo $\equiv_1$, ***VAL***($M/\equiv$) is the quotient of ***VAL***(M) modulo $\equiv_2$ and the interpretation $I^{\equiv}$ in $M/\equiv$ of the relation symbols is defined by

$I^{\equiv}(\underline{r})([d_1],\ldots,[d_m]) = [I(\underline{r})(d_1,\ldots,d_m)]$

for every m-ary relation symbol $\underline{r}\in\underline{R}$ and $d_1,\ldots,d_m\in D$.

As in the classical case we can prove a homomorphism theorem connecting the just considered notions. To do this, we have to define the notion of image of a fuzzy structure by a homomorphism.

Definition 2.9. Let M to M' be two fuzzy structures and (h,k) be a homomorphism from M to M'. Then the *image* of M through (h, k) is the fuzzy structure, we indicate with $Im_{(h,k)}(M)$, such that:

- ***Al***($Im_{(h,k)}(M)$) is the algebraic substructure of ***Al***(M') defined in $h(D)$
- ***VAL***($Im_{(h,k)}(M)$) is the algebraic substructure of ***VAL***(M') defined in $k(V)$
- the fuzzy relations in ***Rel***($Im_{(h,k)}(M)$) are the restrictions to $h(D)$ of the fuzzy relations in ***Rel***(M').

Observe that the fuzzy relations in $Im_{(h,k)}(M)$ are well defined since a fuzzy relation r' in ***Rel***($Im_{(h,k)}(M)$) assumes its values in $k(V)$. In fact, since $k \circ I(\underline{r}) = I'(\underline{r}) \circ h$, we have $I'(\underline{r})(h(D^n)) \subseteq k(V)$. Instead, such an argument falls for the weak homomorphisms and therefore there is a difficulty to define the notion of image through a weak homomorphism. This entails that an *homomorphism theorem* is not possible if we refer to these homomorphisms.

Theorem 2.10. The following claims hold true.

i) Let (h, k) be a homomorphism from M to M' and $\equiv_1$, $\equiv_2$ the kernels of h and k, respectively. Then the pair $(\equiv_1,\equiv_2)$ is a congruence of M we call *the kernel* of (h, k). Moreover if we denote by $\equiv$ such a congruence, the quotient $M/\equiv$ is isomorphic with $Im_{(h,k)}(M)$.

ii) Let $\equiv$ be a congruence in M and $M/\equiv$ be the related quotient. Let h: $\boldsymbol{Al}(M)\to\boldsymbol{Al}(M)/\equiv_1$ and k : $\boldsymbol{VAL}(M)\to \boldsymbol{VAL}(M)/\equiv_2$ be the canonical epimorphisms. Then (h, k) is an epimorphism, we call the *canonical epimorphism*, from M to $M/\equiv$, and $\equiv$ is the kernel of (h,k).

Proof. To prove the first claim, let $\underline{r}$ be a relation symbol, and $d_1,\dots,d_m,b_1,...,b_m$ elements in D such that $d_1\equiv_1 b_1,....,d_m\equiv_1 b_m$. Then $h(d_1) = h(b_1),\dots,h(d_m) = h(b_m)$ and therefore,

$$k(I(\underline{r})(d_1,\dots,d_m)) = I'(\underline{r})(h(d_1),\dots,h(d_m)) = I'(\underline{r})(h(b_1),\dots,h(b_m)) = k(I(\underline{r})(b_1,\dots,b_m))$$

Then $I(\underline{r})(d_1,\dots,d_m) \equiv_2 I(\underline{r})(b_1,\dots,b_m)$ and this proves that $(\equiv_1,\equiv_2)$ is a congruence. It is evident that the maps h': $D/\equiv_1\to h(D)$ and k' : $V/\equiv_2 \to k(V)$ defined by setting $h'([x]) = h(x)$ and $k'([x]) = k(x)$ defines an isomorphism between $M/\equiv$ and $Im_{(h,k)}(M)$. The proof of the second claim is matter of routine.

Observe that the order relation induced in $\boldsymbol{VAL}(M/\equiv)$ by the meet operator is defined by setting

$$[\lambda] \le [\mu] \Leftrightarrow [\lambda]\wedge[\mu] = [\lambda] \Leftrightarrow [\lambda\wedge\mu] = [\lambda] \Leftrightarrow \lambda\wedge\mu \equiv \lambda.$$

This means that the canonical homomorphism is order-preserving.

3. Products and ultraproducts

In this section we will introduce the basic notions of product and ultraproduct of a family of fuzzy models. As usual, if $(S_i)_{i\in I}$ is a family of algebraic structures, then we denote by $\Pi_{i\in I} S_i$ the related direct product. If $(M_i)_{i\in I}$ is a family of fuzzy models such that all the valuation structures $\boldsymbol{VAL}(M_i)$ are complete, then the valuation structure $\Pi_{i\in I}$ $\boldsymbol{VAL}(M_i)$ of its product M is complete.

Definition 3.1. Let $(M_i)_{i\in I}$ be a family of fuzzy models, then we define the *Cartesian product* of $(M_i)_{i\in I}$ as the fuzzy model $M = \Pi_{i\in I} M_i$ such that

$$\boldsymbol{Al}(M) = \Pi_{i\in I}\ \boldsymbol{AL}(M_i),\quad \boldsymbol{VAL}(M) = \Pi_{i\in I}\ \boldsymbol{VAL}(M_i)$$

and, for every n-ary predicate symbol $\underline{r}$ and $f_1,\dots,f_n$ in the domain D of ***Al***(M),

$$I(\underline{r})(f_1,\dots,f_n) = <I_i(\underline{r})(f_1(i),\dots,f_n(i)>_{i\in I} .$$

Then M is defined in the domain $\Pi_{i\in I} D_i$ by the interpretation I such that

$I(\underline{\lambda}) = <I_i(\underline{\lambda})>_{i\in I}$

$I(\underline{o})(\lambda_1,\dots,\lambda_n) = <I_i(\underline{o})(\lambda_1,\dots,\lambda_n)>_{i\in I}$

$I(\underline{c}) = <I_i(\underline{c})>_{i\in I}$

$I(\underline{h})(f_1,\dots,f_m) = <I_i(\underline{h})(f_1(i),\dots,f_n(i))>_{i\in I}$

$I(\underline{r})(f_1,\dots,f_n) = <I_i(\underline{r})(f_1(i),\dots,f_n(i)>_{i\in I}.$

Notice that such a definition is not an extension of the classical one. Indeed if we assume that all the valuation structures ***VAL***(M_i) coincides with the two elements Boolean algebra {0,1}, then the Cartesian product M is not a classical structure since its predicate are evaluated in the Boolean algebra $\Pi_{i\in I}$ ***VAL***(M_i) = $\{0,1\}^I$.

To define the notion of ultraproduct, we need the following proposition.

Proposition 3.2. Let $(M_i)_{i\in I}$ be a family of fuzzy models and let $\mathcal{U}$ be an ultrafilter in $P(I)$. Let $\equiv_1$ and $\equiv_2$ be the congruences defined by $\mathcal{U}$ in the structures $\Pi_{i\in I}$ ***AL***(M_i) and $\Pi_{i\in I}$ ***VAL***(M_i) respectively. Then the pair $(\equiv_1, \equiv_2)$ is a congruence of the Cartesian product $M = \Pi_{i\in I} M_{i\in I}$.

Proof. We observe only that

$$f_1 \equiv_1 g_1 ,\dots,f_n \equiv_1 g_n \Rightarrow \{i\in I : f_1(i) = g_1(i)\}\in \mathcal{U},\dots \{i\in I : f_n(i) = g_n(i)\}\in \mathcal{U}$$

$$\Rightarrow \{i\in I : f_1(i) = g_1(i)\}\cap\dots\cap\{i\in I : f_n(i) = g_n(i)\}\in \mathcal{U}$$

$$\Rightarrow \{i\in I : I_i(\underline{r})(f_1(i),\dots,f_n(i)) = I_i(\underline{r})(g_1(i),\dots,g_n(i))\} \in \mathcal{U}$$

$$\Rightarrow I(\underline{r})(f_1,\dots,f_n) \equiv_2 I(\underline{r})(g_1,\dots,g_n).$$

Definition 3.3. Let $(M_i)_{i\in I}$ be a family of fuzzy models and $\mathcal{U}$ be an ultrafilter in $P(I)$. Then the *ultraproduct* of $(M_i)_{i\in I}$ *modulo* $\mathcal{U}$ is the fuzzy structure $M^{\mathcal{U}} = \Pi^{\mathcal{U}}_{i\in I} M_i$ obtained as the quotient of $\Pi_{i\in I} M_i$ modulo the congruence $(\equiv_1, \equiv_2)$ associated with $\mathcal{U}$.

Then $M^{\mathcal{U}}$ is defined in the domain $(\Pi_{i\in I} D_i)/\equiv_1$ by the interpretation $I^{\mathcal{U}}$ such that

$I^u(\underline{\lambda}) = [<I_i(\underline{\lambda})>_{i\in I}]$

$I^u(\underline{o})([\lambda_1],\dots,[\lambda_n]) = [<I_i(\underline{o})(\lambda_1,...,\lambda_n)>_{i\in I}]$

$I^u(\underline{c}) = [<I_i(\underline{c})>_{i\in I}]$

$I^u(\underline{h})([f_1],\dots,[f_m]) = [<I_i(\underline{h})(f_1(i),...,f_n(i))>_{i\in I}]$

$I^u(\underline{r})(f_1,...,f_n) = [<I_i(\underline{r})(f_1(i),\dots,f_n(i)>_{i\in I}]$.

Note 1. Differently from the case of the product, the ultraproduct of a family of classical models is a classical model, too. Indeed the quotient of $\Pi_{i\in I}$ $\boldsymbol{VAL}(M_i) = \{0,1\}^I$ modulo $\equiv_2$ is the Boolean algebra {0,1}.

Note 2. Notice also that in the classical definition of ultraproduct the ultrafilter does not define a congruence in the direct product. More precisely, the "almost everywhere equal" relation is not compatible with the relations represented in the language. This since in the classical case the direct product is forced to be a model evaluated in the Boolean algebra {0,1} and not in the Boolean algebra $\{0,1\}^I$.

4. Deduction apparatus for fuzzy logic

In all the logics, the deduction apparatus is a tool to elaborate pieces of information and in fuzzy logic, a piece of information is represented by a fuzzy subset.

Let s be a finite fuzzy theory whose support is $\{\alpha_1,...,\alpha_n\}$. Then we can represent s by the finite set $\{(\alpha_1,\lambda_1),..., (\alpha_n,\lambda_n)\}$ of fuzzy formulas where $\lambda_i = s(\alpha_i)$. Equivalently, we represent s by a list as

α_1 $[\lambda_1]$

. . .

α_n $[\lambda_n]$

We can improve the available information s by the proofs and, in turn, this requires a notion of fuzzy inference rule. The following are examples of "fuzzyfication" of three famous classical rules (modus ponens, particularization, and $\wedge$-introduction rule):

$$\langle \frac{\alpha \quad \alpha\rightarrow\beta}{\beta} \qquad \frac{\lambda \quad \mu}{\lambda\otimes\mu} \rangle \quad ;$$

$$\langle \frac{\forall x\alpha}{\alpha(t)} \quad \frac{\lambda}{\lambda}\rangle \ ;$$

$$\langle \frac{\alpha \quad \beta}{\alpha \wedge \beta} \quad \frac{\lambda \quad \mu}{\lambda \otimes \mu}\rangle$$

where t is any term, α and β are formulas of the language, λ, μ elements in the valuation set V and ⊗ is a suitable binary operation. Notice that the values λ and μ in the inference rules are not intended as truth values but as constraints on the possible truth values. So, the operation ⊗ expresses a way to calculate a constraint on the truth value of the conclusion from constraints on the truth values of the premises. More in particular, the meaning of these rules is the following.

Extended modus ponens:

if it is proved that the truth value of α is at least λ and that the truth value of $\alpha\to\beta$ is at least μ, then we can claim that the truth value of β is at least $\lambda\otimes\mu$.

Extended particularization:

if it is proved that the truth value of $\forall x\alpha$ is at least λ and t is a term, then the truth value of $\alpha(t)$ is at least λ.

Extended $\wedge$-introduction rule:

if it is proved that the truth value of α is at least λ and that the truth value of β is at least μ, then we can claim that the truth value of $\alpha\wedge\beta$ is at least $\lambda\otimes\mu$.

When not differently specified, we assume that the valuation structure is a residuated lattice $(V, \otimes , \to, \leq, 0, 1)$ and that the fuzzy inference rules are defined by the product ⊗ in such a lattice. Also, we assume that ⊗ satisfies the continuity condition, i.e. $x\otimes(Sup_{i\in I}x_i) = Sup_{i\in I}\, x\otimes x_i$ for every family $(x_i)_{i\in I}$ of elements in V. Important examples are obtained by setting V equal to the interval [0,1] and by assuming that ⊗ is a continuous triangular norm.

More in general a *fuzzy inference rule* is defined as a pair $r = (r_{syntax}, r_{semantics})$ where r_{syntax} is a partial n-ary operation defined in the set of sentences (i.e. an inference rule in the usual sense) and $r_{semantics}$ is an n-ary operation in the set V of truth values satisfying a continuity property.

Definition 4.1. Let L be a first order language, then a *fuzzy deduction apparatus* in L is a pair (IR, la) where IR is a set of fuzzy inference rules and $la : Form(L) \to V$ is a fixed fuzzy set of formulas we call the *fuzzy subset of logical axioms.*

Notice that, as in classical logic, the fuzzy subset of logical axiom is fixed in a fuzzy logic while the fuzzy subset of proper axioms varies. In classical logic given a set T of hypothesis a proof is a finite sequence of formulas such that every formula is either a logical axiom, or an hypothesis, or is obtained by an inference rule from early proved formulas. Instead in fuzzy logic we can admit every formula as a logical axiom or as an hypothesis. The valuation of the correctness degree of the proof depends on the correctness degree of these assumptions.

Definition 4.2. A *proof π of a formula α* is any sequence $\pi_1,..., \pi_m$ of formulas such that $\pi_m = \alpha$, 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 have to indicate also the rule and the formulas from $\alpha_1,..., \alpha_{i-1}$ used to obtain α_i).

The importance of the justifications is that they are necessary to define the validity degree of the proof. Such a definition is by induction on the length of π (see [17], [28]). Observe that, 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 $\pi(i)$.

Definition 4.3. Given a fuzzy theory s and a proof π, the *valuation $Val(\pi,s)$ of π with respect to s* is defined by induction on the length m of π in accordance with the following rules:

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

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

$Val(\pi,s) = r_{semantics}(Val(\pi(i(1)),s),\ldots,Val(\pi(i(n)),s))$ if α_m is obtained by the rule $r = (r_{syntax}, r_{semantics})$,

and $\alpha_m = r_{syntax}(\alpha_{i(1)},\ldots, \alpha_{i(n)})$ with $1\leq i(1)<m,\ldots,1\leq i(n)<m$).

Notice 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 a hypothesis. So, the first two lines of the definition of 4.3 gives also the induction basis. The value $Val(\pi,s)$ is interpreted as a piece of information on the truth value of α, more precisely, the information: “ *the truth value if α is at least $Val(\pi,s)$*”. Different proofs of the same formula α give different pieces of information on the truth value of α. This suggests the following definition.

Definition 4.4. Given a fuzzy deduction apparatus (*IR*, *la*), the *deduction operator* is the operator $D : V^{Form(L)} \to V^{Form(L)}$ such that, for every $s \in V^{Form(L)}$ the fuzzy subset $D(s)$ is defined by setting,

$$D(s)(\alpha) = Sup\{Val(\pi,s) : \pi \text{ is a proof of } \alpha\} \qquad (4.1)$$

for every formula α.

We emphasize that $D(s)(\alpha)$ represents the best possible information on α we can draw from *s*.

CHAPTER 3

FUZZY LOGIC PROGRAMMING

1. Fuzzy logic programming

Fuzzy logic programming (see [9]) is a very promising section of fuzzy logic whose aim is to build up intelligent data-base systems with "flexible" answers, expert systems able to consider vague predicates and so on, combining the might of logic programming (see [24]) and the big adaptability of fuzzy logic.

We introduce some basic definitions in fuzzy logic programming; observe that in this section L will be a residuated lattice.

Consider a fuzzy deduction apparatus (IR, la) and let s be a *fuzzy subset* of S, we call *support* of s the subset $Supp(s) = \{x \in S : s(x) \neq 0\}$.

As usual, if $\mathcal{L}$ is a first order language, then we denote by $F=Form(\mathcal{L})$ the set of formulas of $\mathcal{L}$ and by $B_{\mathcal{L}}$ the Herbrand base, i.e. the set of facts.

Definition 1.1. A (*positive*) *implicative clause* is either an atomic formula or a formula like $h(\alpha_1,...,\alpha_n)\rightarrow\alpha$ where α, $\alpha_1,\ldots,\alpha_n$ are atomic formulas and $h(\alpha_1,...,\alpha_n)$ is composed only by conjunctions and disjunctions interpreted by continuous norms and co-norms, respectively ($Cl(\mathcal{L})$ is the related set of positive clauses).

Definition 1.2. A fuzzy subset p: $F\rightarrow L$ of formulas is a (*positive*, *ground*) *fuzzy program* if $Supp(p)$ is a set of (positive, ground) implicative clauses.

The following definition of least fuzzy Herbrand model is syntactical in nature. It is possible to prove that such a definition is equivalent with the usual one, semantic in nature (see [18]).

Definition 1.3. Let p be a fuzzy program, we call *least fuzzy Herbrand model* of p the fuzzy subset of facts m_p we can derive from p, i.e. the restriction of $D(p)$ to $B_{\mathcal{L}}$ (where D is the deduction operator define as in definition 4.4., in chapter 2)

Such a notion depends on the considered deduction apparatus, obviously.

Denote by $Gr(p)$ the set of ground instances of the formulas in p and by $Fact(p)$ the set of facts in $Gr(p)$. Then, *the one-step consequence operator* is the function $T_P : P(B_L) \to P(B_L)$ defined by setting

$$T_P(S) = \{\alpha \in B_L : \alpha \leftarrow \alpha_1 \wedge ... \wedge \alpha_m \in Gr(p),\ \alpha_1 \in S, ..., \alpha_m \in S\} \cup Fact(p)$$

for every subset S of B_L.

Such an operator enables us ([18]) to obtain the least Herbrand model m_P of p by the equation

$$m_P = \cup_{n\in N} T_P{}^n(\emptyset). \tag{1.1}$$

Observe that equation (1.1) 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 [2] and [19], then, under very natural hypotheses, it is easy to show that m_p is a recursively enumerable fuzzy subset of B_L.

A consequence is the following theorem (see [18]) that shows that the least fuzzy Herbrand model of a positive fuzzy program represents the informative content of p.

Theorem 1.4. 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,

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

Observe that as in the classical case, several difficulties exist for fuzzy programs which are not positive.

2. Fuzzy logic meta-programming

The idea is to extend fuzzy logic programming to take into account the synonymy relation among predicates in accordance with the similarity logic proposed by M. S. Ying in [33]. The idea of Ying is that it is possible to relax the application of the inference rules in such a way that it is also admitted an approximate matching of the predicate names. As an example it is admitted that from α and $\alpha' \to \beta$ we can infer β even in the case that α' is only approximately equal to α. An application to such an idea to logic programming was done in several papers (see [1], [3], [5], [11]) where the definition of synonymy refers to Gödel's norm. Successively, in [25] it's shows that it is possible to define a similarity logic programming (in particular a synonymy logic programming) in the framework of multi-adjoint logic programming. The proposed procedure works with any triangular norm and the authors show that the resulting logic coincides with the existing ones in the case of Gödel's norm.

Since the synonymy is a meta-relation, in order to define a synonymy logic we have to consider a suitable meta-logic. On the other hand, all the definitions in fuzzy logic programming and in similarity logic programming can be expressed by positive clauses in classical logic programming. Then, we show that given a fuzzy program in a language L, we can translate it into an equivalent classical program in a suitable (meta-) language L_m. Since the predicate names in L become constants in L_m, this enables us to admit in L_m meta-relations among predicates. In particular, the meta-relation the paper is interested is the synonymy and this enable us to define a synonymy-sensitive fuzzy logic programming.

There are at least three reasons in favour of such a logic. The first one is that, differently from the papers [1], [3], [11], all the triangular norms are admitted. The second is that the resulting notion of fuzzy Herbrand model is uniformly continuous with respect to the synonymy relation (a basic property for a synonymy logic). Finally, another reason is that the resulting logic is a similarity logic in the abstract sense given in [17]. This means that its deduction operator is the closure operator obtained by combining the similarity closure operator with the one-step consequence operator associated with the given fuzzy program.

3. Translation of a classical program

As a first step, we consider a way to translate a classical program (in a language L) into another simple classical program (in a suitable "meta-language" L_m). Successively we will extend it to the fuzzy case. As an example, consider the following program *P*

$r(a,b)$

$r(b,c)$

$r(c,d)$

$r(s,s)$

$sr(X,Y) \leftarrow r(Y,X)$

$sr(X,Y) \leftarrow r(X,Y)$.

The language L of *P* has two predicate symbols *r* and *sr* and the constants *a*, *b*, *c*, *d*, *s*. Now, we can interpret the fact $r(a,b)$ in a meta-linguistic level by claiming that:

the sentence " the relation r is satisfied by a and b" is an axiom.

Likewise, we can interpret the instance $sr(b,a) \leftarrow r(a,b)$ of the rule $sr(X,Y) \leftarrow r(Y,X)$ by claiming

if the sentence " the relation r is satisfied by a and b" is a theorem,

then the sentence "the relation sr is satisfied by b and a" is a theorem.

Then we have to consider a language L_m in which, by a reification process, the predicate symbols *r* and *sr* become two constants and in which there are two predicates corresponding to the notions "*to be an axiom*" and "*to be a theorem*". In such a language we can translate the program *P* into the following program

$ax(r,a,b)$

$ax(r,b,c)$

$ax(r,c,d)$

$ax(r,s,s)$

$th(sr,X,Y) \leftarrow th(r,X,Y)$

$th(sr,X,Y) \leftarrow th(r,Y,X)$.

Obviously, we have to add also a general rule to claim that any axiom is a theorem

$th(R, X, Z) \leftarrow ax(R, X, Z)$.

We can also avoid such a rule and to substitute directly *ax* with *th*. Notice that we cannot interpret $th(sr,X,Y) \leftarrow th(r,Y,X)$ as "if $r(Y,X)$ is a theorem then $r(X,Y)$ is a theorem" since this should be the interpretation of the formula $\forall X\forall Y(sr(X,Y)) \leftarrow (\forall X\forall Y\ (r(Y,X)))$. Instead a ground instance as $th(sr,b,a) \leftarrow th(r,a,b)$ of such a rule is correctly interpreted as "if $r(a,b)$ is a theorem then $r(b,a)$ is a theorem".

Definition 3.1. Given a first order language L we denote by L_m the language such that:

- the constants of L_m are obtained by adding to the constants of L all the predicate symbols of L
- the function symbols are the same as in L
- there is a predicate symbol th_n of arity n+1 for every arity n of a predicate symbol in L.

A translation function from L to L_m is defined as follows.

Definition 3.2. The *translation function* is the map τ: $Form(L) \rightarrow Form(L_m)$ defined by setting

$\tau(r(t_1,\ldots,t_n)) = th_n(r,t_1,\ldots,t_n)$.

$\tau(\alpha_1\wedge\alpha_2) = \tau(\alpha_1)\wedge\tau(\alpha_2)$,

$\tau(\alpha_1\vee\alpha_2) = \tau(\alpha_1)\vee\tau(\alpha_2)$,

$\tau(\neg\alpha) = \neg\tau(\alpha)$,

$\tau(\forall x_i\alpha) = \forall x_i\tau(\alpha)$.

In particular, the translation of a (positive) clause $\alpha \leftarrow\alpha_1\wedge\ldots\wedge\alpha_h$ is the (positive) clause $\tau(\alpha) \leftarrow\tau(\alpha_1)\wedge\ldots\wedge\tau(\alpha_h)$. This means that if P is a (positive) program, then $\tau(P)$ is a (positive) program, too.

To simplify our notations, in the sequel we will write $th(r,t_1,\ldots,t_n)$ instead of $th_n(r,t_1,\ldots,t_n)$. Equivalently, we can consider only a monadic predicate *th* in L_m and some way to represent a vector $(x_0,x_1,...,x_n)$. For example, we can add in L_m a name *List* for a binary function, and to write $(x_0,x_1,...,x_n)$ to denote the term $List((x_0,x_1,...,x_{n-1}),x_n)$ and (x_0) to denote x_0. Then the translation can be defined by substituting the first rule with

$\tau(r(t_1,\ldots,t_n)) = th((r,t_1,\ldots,t_n))$.

It is evident that,

$T \,|\!\!- \alpha \Leftrightarrow \tau(T) \,|\!\!- \tau(\alpha)$

where |- is the classical entailment relation, T is a theory and α a formula. To show this it is sufficient to observe that:

- if Ax is the set of logical axioms, then $\alpha \in Ax$ if and only if $\tau(\alpha) \in Ax$
- the translation is compatible with the inference rules.

In particular, if P is a program and α a fact, $\alpha \in M_P \Leftrightarrow \tau(\alpha) \in M_{\tau(P)}$

where M_P and $M_{\tau(P)}$ are the Herbrand models of P and $\tau(P)$, respectively. We give a step-by-step proof of such an equivalence in the perspective of its extension to fuzzy logic programming.

Theorem 3.3. Let P be a positive program and let M_P and $M_{\tau(P)}$ be the Herbrand models of P and $\tau(P)$, respectively. Then,

$$\alpha \in M_P \Leftrightarrow \tau(\alpha) \in M_{\tau(P)}. \qquad (3.1)$$

Proof. It is not restrictive to assume that P is ground. To prove (3.1) it is sufficient to prove that, for every fact α and $n \in N$,

$$\alpha \in T_P{}^n(\emptyset) \Leftrightarrow \tau(\alpha) \in T_{\tau(P)}{}^n(\emptyset). \qquad (3.2)$$

We will prove this by induction on n. Indeed, in the case n =1, since $T_P(\emptyset) = Fact(P)$ and $T_{\tau(P)}(\emptyset) = Fact(\tau(P)) = \tau(Fact(P))$, (3.2) is evident. Consider the case $n \neq 1$, assume that (2.2) is satisfied by n-1 and that $\alpha \in T_P{}^n(\emptyset) = T_P(T_P{}^{n-1}(\emptyset))$. Then either $\alpha \in T_P{}^{n-1}(\emptyset)$ or there is a rule $\alpha \leftarrow \alpha_1 \wedge \ldots \wedge \alpha_h$ in P such that $\alpha_1 \in T_P{}^{n-1}(\emptyset),\ldots,\alpha_h \in T_P{}^{n-1}(\emptyset)$. In the first case, by the induction hypothesis $\tau(\alpha) \in T_P{}^{n-1}(\emptyset)$ and therefore $\tau(\alpha) \in T_P{}^n(\emptyset)$. In the latter, by the induction hypothesis $\tau(\alpha_1) \in T_{\tau(P)}{}^{n-1}(\emptyset),\ldots, \tau(\alpha_h) \in T_{\tau(P)}{}^{n-1}(\emptyset)$ and, since the rule $\tau(\alpha) \leftarrow \tau(\alpha_1) \wedge \ldots \wedge \tau(\alpha_h)$ is in $\tau(P)$, this entails that $\tau(\alpha) \in T_{\tau(P)}{}^n(\emptyset)$.

Conversely, assume that $\tau(\alpha) \in T_{\tau(P)}{}^n(\emptyset) = T_{\tau(P)}(T_{\tau(P)}{}^{n-1}(\emptyset))$. Then either $\tau(\alpha) \in T_{\tau(P)}{}^{n-1}(\emptyset)$ or there is a rule $\beta_1 \wedge \ldots \wedge \beta_h \rightarrow \beta$ with $\beta = \tau(\alpha)$ in $\tau(P)$ such that

$\beta_1 \in T_{\tau(P)}{}^{n-1}(\emptyset),\ldots,\beta_h \in T_{\tau(P)}{}^{n-1}(\emptyset)$. In the first case by induction hypothesis

$\alpha \in T_P{}^{n-1}(\emptyset)$ and therefore $\alpha \in T_P{}^n(\emptyset)$. In the latter, let $\alpha' \leftarrow \alpha_1 \wedge \ldots \wedge \alpha_h$ be a rule in P whose translation is $\beta \leftarrow \beta_1 \wedge \ldots \wedge \beta_h$. Then

$\tau(\alpha') = \beta = \tau(\alpha)$, $\tau(\alpha_1) = \beta_1 \in T_{\tau(P)}{}^{n-1}(\emptyset), \ldots, \tau(\alpha_h) = \beta_h \in T_{\tau(P)}{}^{n-1}(\emptyset)$.

Since by inductive hypothesis, $\alpha_1 \in T_P{}^{n-1}(\emptyset),\ldots,\alpha_h \in T_P{}^{n-1}(\emptyset)$, this entails that $\alpha' \in T_P{}^n(\emptyset)$. Since τ is injective, we can conclude that $\alpha \in T_P{}^n(\emptyset)$.

The translation of a program P into the meta-program $\tau(P)$ is proposed as a first step towards a possible translation of a fuzzy logic program into a classical meta-program. Nevertheless, perhaps this translation gives some advantages also in the case we confine ourselves to classical logic programming. As an example, if we admit the meta-predicate "*is the symmetric extension of*", then we can consider the following translation of the proposed example:

$th(r,a,b)$

$th(r,b,c)$

$th(r,c,d)$

$th(r,s,s)$

$symm\_exten(sr,r)$

$th(R_2,X,Y) \leftarrow symm\_exten(R_2,R_1) \wedge th(R_1,X,Y)$

$th(R_2,X,Y) \leftarrow symm\_exten(R_2,R_1) \wedge th(R_1,Y,X))$.

The advantage of such a translation is that the two meta-rules give a general procedure for the symmetric extension of a relation. So we can add such a procedure to our library. In a similar uniform way we can define, for example, procedures for the reflexive extension and the transitive extension of a relation.

Observe that the idea for a translation of classical logic into classical logic programming was examined in literature in a extensive way (see for example [7], [23]).

4. An example of fuzzy logic programming

We refer to a residuated lattice (V, ⊗ , →, ≤, 0, 1) and to a deduction apparatus with no logical axiom and whose fuzzy rules are the extended Modus Ponens, the extended ∧-introduction rule and the extended particularization. As an example, consider the following fuzzy program p in the interval [0,1]:

$loves(carl,luise)$	[0.3]
$loves(carl,mary)$	[0.2]
$loves(carl,X) \leftarrow young(X) \wedge beautiful(X)$	[0.9]
$beautiful(mary)$	[0.8]
$young(mary)$	[0.7]
$young(helen)$	[0.7]

we represent by a set of *fuzzy rules*, i.e. as the set of pair (α, $p(\alpha)$), $p(\alpha) \neq 0$. Also, assume that we will calculate, for example, the value $m_p(loves(carl, mary))$. Then, a simple proof of the fact $loves(carl, mary)$, we denote byπ_1, consists in the observation that such a formula is an axiom at degree 0.2 (i.e. it is true at least at degree 0.2), and this gives the constraint $Val(\pi_1,p) = 0.2$. A different proof π_2 is obtained by observing that, by particularization, we obtain the formula

$loves(carl, mary) \leftarrow young(mary) \wedge beautiful(mary)$

with truth degree 0.9. On the other hand, by the $\wedge$-introduction rule, we obtain the formula

$young(mary) \wedge beautiful(mary)$

with truth degree $0.8 \otimes 0.7$. Afterwards, by Modus Ponens, we get

$loves(carl, mary)$

with truth degree $0.9 \otimes 0.8 \otimes 0.7$. In the case $\otimes$ is the usual product, this gives the value $val(\pi_2,p) = 0.504$. Since there is no further proof for such a fact, we can conclude that

$m_p(loves(carl, mary)) = max\{Val(\pi_1,p), Val(\pi_2,p)\} = 0.504.$

Instead, if we consider the fact $loves(carl, helen)$, then there is no proof for such a fact using the formulas in the support of p. On the other hand, since the fuzzy program p assigns to $loves(carl, helen)$ the value 0, the observation that $loves(carl, helen)$ is an axiom at degree 0 is a proof with degree 0. Then,

$m_p(loves(carl, helen)) = 0.$

We have a general way to calculate the least Herbrand model of a fuzzy program, by extending the fixed point method of classical logic programming.

We recall that the fuzzy subset of ground instances of clauses in p is the fuzzy program $Gr(p)$: $Cl(L) \rightarrow V$ defined by setting $Gr(p)(\alpha)=0$ if α is not ground and

$$Gr(p)(\alpha) = Sup\{p(\underline{\alpha}) : \alpha \text{ is a ground instance of a clause } \underline{\alpha}\},$$

otherwise. The supremum is justified by the fact that it is possible for a formula α to be the ground instance of more than one formula in $Supp(p)$.
As an example if $p(r(x,b)) = 0.7$ and $p(r(a,y)) = 0.5$, then we have to set $Gr(p)(r(a,b)) = max\ \{0.5,0.7\} = 0.7$.

We say that p is *ground* if $Gr(p) = p$. We recall also that the fuzzy subset of fact of p is the restriction of $Gr(p)$ to the Herbrand base, i.e. the fuzzy subset $Fact(p) : B_L \rightarrow V$ defined by

$Fact(p)(\alpha) = Sup\{p(\underline{\alpha}) : \alpha$ is a ground instance of an atomic formula $\underline{\alpha}\}$.

In the case p is ground,

$Fact(p)(\alpha) = p(\alpha).$

We give an equivalent definition for the one-step consequence operator:

Definition 4.1. Let p: $Cl(L) \to V$ be a fuzzy program. Then the *one-step consequence operator* is the operator $T_p : V^{B_L} \to V^{B_L}$ defined by setting, for every $s \in V^{B_L}$,

$$T_p(s) = T^*_p(s) \cup Fact(p)$$

where, in turn $T^*_p : V^{B_L} \to V^{B_L)}$ is defined by setting, for every $\alpha \in B_L$,

$$T^*_p(s)(\alpha) = Sup\{Gr(p)(\alpha \leftarrow \alpha_1 \wedge ... \wedge \alpha_m) \otimes s(\alpha_1) \otimes ... \otimes s(\alpha_m) : \alpha \leftarrow \alpha_1 \wedge ... \wedge \alpha_m \in Supp(Gr(p))\}.$$

Observe that the least fuzzy Herbrand model for a program p, is given by

$$m_p = \cup_{n \in N} T_p^n(\emptyset).$$

and this entails that the fuzzy least Herbrand model of p coincides with the fuzzy least Herbrand model of $Gr(p)$. So, in all the proofs it is not restrictive to assume that p is ground.

5. A meta-logic for fuzzy logic programming

In this section we will show how translate a fuzzy program in a language L into a classical meta-program in a language L_m. To this aim, again we have assume that in L_m there are the predicate names th_n and a constant for every predicate name in L. In addition, since we have to write in an explicit way the involved truth values, in L_m we put constants to denote the truth values. More precisely, since it is not reasonable to admit a language which is not enumerable, it is useful to refer only to a particular class of truth values. Then, in accordance with domain theory, we consider the following definition where $\prec$ is the relation in V defined by setting $b \prec x$ provided that for every nonempty upward directed subset A of V

$$x \leq supA \Rightarrow \text{there is } a \in A \text{ such that } b \leq a.$$

Definition 5.1. We say that a residuated lattice $(V, \leq, \otimes, \to, 0, 1)$ is a *continuous residuated lattice with enumerable basis* $\boldsymbol{B}$, provided that $\otimes$ is continuous, $\boldsymbol{B}$ is an enumerable sublattice of V closed with respect to $\otimes$ and, for every $x \in V$,

$$x = sup(\{b \in \boldsymbol{B} : b \prec x\})$$

In other words, in a continuous residuated lattice all the elements can be approximate *"from below"* by elements in $\boldsymbol{B}$. As an example, we can consider the case V is the lattice [0,1], $\boldsymbol{B}$ is the set of rational numbers in [0,1] and $\otimes$ is the usual product. In such a case $\prec$ is the strict order. Further example are obtained by assuming that $\otimes$ is one of the triangular norms usually considered in literature. Also, every finite residuated lattice is a continuous residuated lattice provided that we put ***B*** = *V*. In such a case $\prec$ coincides with the order relation ≤. If *S* is an enumerable set, then *P*(*S*) is a continuous residuated lattice in which a basis is defined by the class of all the finite subsets of *S*. In such a case $b \prec x$ if and only if *b* is a finite part of *x*.

In the following we consider only continuous residuated lattices with an enumerable basis ***B*** and fuzzy programs with values in ***B*.** Also, in the meta-language L_m we put only an enumerable amount of constants to denote the elements in ***B***.

Another question is that in the translation, we have to represent in some way the product ⊗. To do this we assume that in L_m there is a predicate *product* and that in the translation we consider the *diagram* of the algebraic structure (***B***,⊗), i.e. the (decidable) set of facts

$$Diagr(\boldsymbol{B}) = \{product(\lambda,\mu,\gamma) : \gamma = \lambda \otimes \mu \,;\, \lambda, \mu, \gamma \in \boldsymbol{B}\}.$$

As an example, the translation of the fuzzy program given in Section 4 is obtained by adding to *Diagr*(***B***) the program

th(*loves*, *carl*, *luise*, 0.3).

th(*loves*, *carl*, *mary*, 0.2).

$th(loves, carl, X, Z) \leftarrow th(young, X, Z_1) \wedge th(beautiful, X, Z_2) \wedge product(Z_1,Z_2,Z_3) \wedge product(Z_3, 0.9, Z)$.

th(*beautiful*, *mary*, 0.8).

th(*young*, *mary*, 0.7).

th(*young*, *helen*, 0.7).

More precisely we have to consider also the default rules

th(*R*, *X*, 0).

th(*R*, *X*, *Y*, 0).

claiming that that every fact in *L* can be proved at least with truth degree 0. Thus, we propose the following general definition.

Definition 5.2. Let *L* be a first order language and let (*V*, ⊗ , →, ≤, 0, 1) be a residuated lattice with an enumerable basis ***B***. Then we denote by L_m the first order language such that

- the constants of L_m are obtained by adding to the constants in *L* all the predicate symbols of *L* and all the elements λ in ***B***
- the function symbols in L_m are the same as in *L*
- in L_m there is a predicate symbol th_n for every *n* which is the arity of a predicate symbol in *L*
- in L_m there is a predicate *product*.

As early argued in Section 3, we can assume also that in L_m there is only a monadic predicate *th*. In the next definition we write $Z = X \otimes Y$ to denote the formula $product(X,Y,Z)$, $Z = X \otimes Y \otimes A$ to denote the formula $product(X,Y,Z1) \wedge product(Z1, A, Z)$ and so on.

Definition 5.3. Consider a first order language L, a continuous residuated lattice $(V, \otimes , \rightarrow, \leq, 0, 1)$ with an enumerable basis $\boldsymbol{B}$ and the corresponding meta-language L_m. Then, given a clause α and a variable Z, we define the formula $\tau(\alpha,Z)$ in L_m by setting

$\tau(r(t_1,\ldots,t_n),Z) = th(r,t_1,\ldots,t_n, Z)$ for every atomic formula $r(t_1,\ldots,t_n)$ in L

$\tau(\alpha \leftarrow \alpha_1 \wedge \ldots \wedge \alpha_n, Z) = \tau(\alpha, Z_{n+1}) \leftarrow \tau(\alpha_1,Z_1) \wedge \ldots \wedge \tau(\alpha_n,Z_n) \wedge (Z_{n+1} = Z_1 \otimes \ldots \otimes Z_n \otimes Z)$

for every rule $\alpha \leftarrow \alpha_1 \wedge \ldots \wedge \alpha_n$ in L and where the variables $Z, Z_1,\ldots,Z_{n+1}$ are pairwise distinct and not occurring in $\alpha, \alpha_1,\ldots,\alpha_n$.

For example,

$\tau(loves(carl, X) \leftarrow young(X) \wedge beautiful(X))=$

$= \tau(loves(carl,X), Z_3) \leftarrow \tau(young(X),Z_1) \wedge \tau(beautiful(X), Z_2) \wedge (Z_3 = Z_1 \otimes Z_2 \otimes Z)=$

$= th(loves, carl, X, Z_3) \leftarrow th(young, X, Z_1) \wedge th(beautiful, X, Z_2) \wedge (Z_3 = Z_1 \otimes Z_2 \otimes Z).$

To translate a program we have to consider the set *Dfl* of default formulas $th(R,X,0)$, $th(R,X,Y,0)$, …, .

Definition 5.4. Consider a fuzzy program *p* in the language L. Then the *translation* of *p* is the classical program $\tau(p)$ in L_m defined by setting

$\tau(p) = \{ \tau(\alpha,Z)_{Z/p(\alpha)} : \alpha \text{ is a positive clause}\} \cup Diagr(\boldsymbol{B}) \cup Dfl$

where $\tau(\alpha,Z)_{Z/p(\alpha)}$ denotes the formula obtained from $\tau(\alpha,Z)$ by substituting in Z the value $p(\alpha)$.

In order to simplify our notation we avoid to write the diagram of valuation structure and the default rules in an explicit way. So the translation of the fuzzy program in Section 4 becomes:

th(*loves*, *carl*, *luise*, 0.3).

th(*loves*, *carl*, *mary*, 0.2).

th(*beautiful*, *mary*, 0.8).

th(*young*,*mary*,0.7).

th(*young*, *helen*,0.7).

$th(loves, carl, X, Z_3) \leftarrow th(young, X, Z_1) \wedge th(beautiful, X, Z_2) \wedge (Z_3 = Z_1 \otimes Z_2 \otimes 0.9)$

In such a case, given the query $th(loves,carl, mary, Z)$, we obtain the answers $Z = 0$, $Z = 0.2$ and $Z = 0.504$. In accordance with the definition of deduction operator, this means that the best constraint on the truth value of the formula $loves(carl,mary)$ is the maximum 0.504.

To prove the equivalence between a fuzzy program p and its translation $\tau(p)$, it is useful the following very interesting lemma given in [32].

Lemma 5.5. Let $(M, \otimes, \leq, 1)$ be a finitely generated ordered monoid. Then every nonempty subset of M admits a maximal element and therefore every nonempty totally ordered subset of M admits a maximum.

As an immediate consequence we obtain the following lemma.

Lemma 5.6. Assume that V is totally ordered and that the fuzzy program p assumes only a finite number of values in $\boldsymbol{B}$. Let $(M, \otimes, \leq, 1)$ be the submonoid of $(\boldsymbol{B},\otimes,\leq,1)$ generated by the values assumed by p and let s be a fuzzy subset of facts assuming its values in M. Then for every fact α there is a rule $\alpha \leftarrow \alpha_1 \wedge ... \wedge \alpha_m$ such that

$$T^*_p(s)(\alpha) = p(\alpha \leftarrow \alpha_1 \wedge ... \wedge \alpha_m) \otimes s(\alpha_1) \otimes ... \otimes s(\alpha_m) \qquad (5.1)$$

Proof. Since

$$\{p(\alpha \leftarrow \alpha_1 \wedge ... \wedge \alpha_m) \otimes s(\alpha_1) \otimes ... \otimes s(\alpha_n) : \alpha \leftarrow \alpha_1 \wedge ... \wedge \alpha_m \in Supp(p)\}$$

is a subset of M, by Lemma 5.5 it admits a maximum max. Then a rule $\alpha \leftarrow \alpha_1 \wedge ... \wedge \alpha_m$ exists such that $max = p(\alpha \leftarrow \alpha_1 \wedge ... \wedge \alpha_m) \otimes s(\alpha_1) \otimes ... \otimes s(\alpha_m)$ and such a rule satisfies (5.1).

Theorem 5.7. Assume that V is totally ordered and that the fuzzy program p assumes only a finite number of truth values. Then, for every fact $r(t_1,\dots,t_k)$,

$$m_p(r(t_1,\dots,t_k)) = Sup\{\lambda \in \boldsymbol{B} : th(r,t_1,\dots,t_k,\lambda) \in M_{\tau(p)}\}. \qquad (5.2)$$

Proof. It is not limitative to assume that p is ground. To prove (5.2) it is sufficient to prove that, for every $\lambda \in \boldsymbol{B}$ and $n \in N$,

$$T_p{}^n(\emptyset)(r(t_1,\dots,t_k)) = \lambda \Leftrightarrow th(r,t_1,\dots,t_k,\lambda) \in T_{\tau(p)}{}^n(\emptyset). \qquad (5.3)$$

Now in the case $n = 1$ observe that $T_p(\emptyset)(r(t_1,\dots,t_k)) = Fact(p)(r(t_1,\dots,t_k)) = p(r(t_1,\dots,t_k))$ and $T_{\tau(p)}(\emptyset) = Fact(\tau(p)) = \tau(Fact(p))$. Then (5.3) follows from the definition of τ. Consider the case $n \neq 1$ and, by induction hypothesis, assume that (5.3) is satisfied by n-1. Then, if $T_p{}^n(\emptyset)(r(t_1,\dots,t_k)) = \lambda$, since

$$\begin{aligned} T_p{}^n(\emptyset)(r(t_1,\dots,t_k)) &= T_p(T_p{}^{n-1}(\emptyset)(r(t_1,\dots,t_k))) \\ &= T^*{}_p(T_p{}^{n-1}(\emptyset))(r(t_1,\dots,t_k)) \vee Fact(p)(r(t_1,\dots,t_k)), \end{aligned}$$

we have to consider two cases. In the case $\lambda = Fact(p)(r(t_1,\dots,t_k))$, it is evident that $th(r,t_1,\dots,t_k,\lambda)\in Fact(\tau(p)) \subseteq T_{\tau(p)}{}^{n}(\emptyset)$. In the case $\lambda = T^{*}{}_{p}(T_p{}^{n\text{-}1}(\emptyset))(r(t_1,\dots,t_k))$, by Lemma 5.6 there is a fuzzy inference rule in p,

$$r(t_1,\dots,t_k) \leftarrow r_1(t_{1,1}, .., t_{1,k(1)}) \wedge \dots \wedge r_m(t_{m,1}, .., t_{m,k(m)}) \qquad [\mu],$$

such that,

$$\lambda = \mu \otimes T_p{}^{n\text{-}1}(\emptyset)(r_1(t_{1,1},.., t_{1,k(1)})) \otimes ... \otimes T_p{}^{n\text{-}1}(\emptyset)(r_m(t_{m,1},.., t_{m,k(m)})).$$

Set $\lambda_i = T_p{}^{n\text{-}1}(\emptyset)(r_i(t_{m,1}, .., t_{i,k(i)}))$, then, since by induction hypothesis

$$th(r_i, t_{i,1}, .., t_{i,k(i)}, \lambda_i) \in T_{\tau(p)}{}^{n\text{-}1}(\emptyset) \text{ and}$$

$$th(r,t_1,\dots,t_k,\lambda) \leftarrow th(r_1,t_{1,1},..,t_{1,k(1)},\lambda_1) \wedge \dots \wedge th(r_m,t_{m,1},..,t_{m,k(m)},\lambda_m)) \wedge$$

$$\wedge\ (\lambda = \lambda_1 \otimes \dots \otimes \lambda_m \otimes \mu),$$

is a ground instance of a rule in $\tau(p)$, we can conclude that $th(r,t_1,\dots,t_k, \lambda) \in T_{\tau(p)}{}^{n}(\emptyset)$.

Conversely, assume that $th(r,t_1,\dots,t_k, \lambda) \in T_{\tau(p)}{}^{n}(\emptyset)$ and that $th(r,t_1,\dots,t_k, \lambda)$ is obtained from $T_{\tau(p)}{}^{n\text{-}1}(\emptyset)$ by the rule

$$th(r,t_1,\dots,t_k, Z_{n+1}) \leftarrow t(r_1, t_{1,1}, .., t_{1,k(1)}, Z_1) \wedge \dots \wedge th(r_m, t_{m,1}, .., t_{m,k(m)}, Z_m)) \wedge$$

$$\wedge\ (Z_{n+1} = Z_1 \otimes \dots \otimes Z_m \otimes \mu),$$

Then there is a ground instance

$$th(r,t_1,\dots,t_k, \lambda) \leftarrow th(r_1,t_{1,1},..,t_{1,k(1)},\lambda_1) \wedge \dots \wedge th(r_m, t_{m,1},.., t_{m,k(m)}, \lambda_m)) \wedge$$

$$\wedge\ (\lambda = \lambda_1 \otimes \dots \otimes \lambda_m \otimes \mu),$$

of such a rule such that

$$th(r_1, t_{1,1},.., t_{1,k(1)}, \lambda_1) \in T_{\tau(p)}{}^{n\text{-}1}(\emptyset), \dots, th(r_m, t_{m,1},.., t_{m,k(m)}, \lambda_m)) \in T_{\tau(p)}{}^{n\text{-}1}(\emptyset).$$

Let

$$r(t_1,\dots,t_k) \leftarrow r_1(t_{1,1},..,t_{1,k(1)}) \wedge \dots \wedge r_m(t_{m,1},.., t_{m,k(m)}) \qquad [\mu],$$

be the fuzzy rule in the fuzzy program p whose translation coincides with the considered rule in $\tau(p)$. Then, since by inductive hypothesis,

$$T_p{}^{n\text{-}1}(\emptyset)(r_1, t_{1,1},.., t_{1,k(1)}) = \lambda_1, \ ... \ , T_p{}^{n\text{-}1}(\emptyset)(r_m, t_{m,1},.., t_{m,k(m)}) = \lambda_m$$

we can conclude that

$$T_p{}^{n}(\emptyset)(r(t_1,\dots,t_k)) = \mu \otimes T_p{}^{n\text{-}1}(\emptyset)(r_1,t_{1,1},..,t_{1,k(1)}) \otimes ... \otimes T_p{}^{n\text{-}1}(\emptyset)(r_m,t_{m,1},..,t_{m,k(m)})) = \lambda.$$

6. Similarity logic and synonymy

We are now ready to face the main question we are interested. We start from an example. Let us suppose that a bookshop assistant have a request, by a customer, for an adventurous and economic book. Moreover let us suppose that he doesn't find any book with these characteristics, then he tries to recommend a book which is sufficiently close to the customer's request. For example he could propose a fantasy book which is not expensive. This attitude is a typical use of synonymy in reasoning in everyday life. Now, it is evident that the available information is vague in nature and therefore that we have to represent it by a fuzzy set of claims as

if x is adventurous and economic then x is good (at degree 1)

"I Robot" is a fantasy story (at degree 0.6)

"I Robot" is not expensive (at degree 1)

"adventurous" is a synonymous of "fantasy" (at degree 0.8)

"economic" is synonymous of "not expensive" (at degree 0.7)

More formally

$Adventurous(x) \wedge Economic(x) \Rightarrow Good(x)$ (at degree 1)

Fantasy(*"I_robot"*) (at degree 0.6)

Not_expensive(*"I_robot"*) (at degree 1)

synonymous(*"adventurous"*, *"fantasy"*) (at degree 0.8)

synonymous(*"economic"*, *"not_expensive"*) (at degree 0.7)

Unfortunately, we cannot consider such a list of fuzzy formulas in the framework of first order fuzzy logic. Indeed, there are words as *adventurous*, *fantasy*, *economic*, *not_ expensive* occurring both as predicates symbols and as constants. In a series of papers (see, for example, [11]) such a question was faced by relaxing the notion of matching between predicates. This means that one considers only the first order fuzzy formulas

$Adventurous(x) \wedge Economic(x) \Rightarrow Good(x)$ [1]

Fantasy(*"I_robot"*) [0.6]

Not_expensive(*"I_robot"*) [1]

while the information about the synonymy between predicates is used to calculate the degree of admissibility of an approximate matching.

In this paper we propose a different approach in which we simply add to the meta-language L_m the predicate symbol "*synonymous*". In accordance, we formalize the information in the considered example by the program

th(*good*, *X*, *Z*) ← *th*(*adventurous*, *X*, V_1)∧*th*(*economic*, *X*, V_2)∧($Z = V_1 \otimes V_2$).

th(*fantasy, I_robot*, 0.6).

th(*notexpensive*, *I_robot*, 1).

synonymous(*adventurous*, *fantasy*, 0.8).

synonymous(*economic*, *notexpensive*, 0.7).

th(*A*,*X*,*V*) ← *synonymous*(*A*, *A'*, V_1)∧*th*(*A*, *X*, V_2)∧($V = V_1 \otimes V_2$).

As usual in fuzzy logic, the intended meaning of a fact as *synonymous*(*economic*, *notexpensive*, 0.7) is that *economic* is a synonymous of *notexpensive* at degree at last 0.7. Equivalently, we can consider the predicate symbol *synonymous* as a constant and to consider the program

th(*good*, *X*, *Z*) ← *th*(*adventurous*, *X*, V_1)∧*th*(*economic*, *X*, V_2)∧($Z = V_1 \otimes V_2$).

th(*fantasy*, *I_robot*,0.6).

th(*notexpensive*, *I_robot*, 1).

th(*synonymous*, *adventurous*, *fantasy*, 0.8).

th(*synonymous*, *economic*, *notexpensive*,0.7).

th(*A*,*X*,*V*) ← *th*(*synonymous*, *A*, *A'*, V_1)∧*th*(*A'*, *X*, V_2)∧($V = V_1 \otimes V_2$).

Given such a program and, for example, the query *th*(*good*, *I_robot*, *Z*), we obtain the answer $Z = 1 \otimes 0.8 \otimes 0.6 \otimes 0.7 \otimes 1$. Notice that such a program is not complete since it is natural to assume that a synonymy satisfies suitable properties, namely that it is a similarity.

Definition 6.1. Given a first order language L, we call ⊗-*synonymy*, in brief *synonymy*, any ⊗-similarity *syn* on the set of predicate symbols such that *syn*(*r*,*r'*) = 0 for every pair of predicate symbols *r* and *r'* with different arities.

It is evident that a synonymy is a fuzzy model of a suitable fuzzy program. Then, it is useful to represent it by a (small) fuzzy set of facts and suitable rules corresponding to three properties (reflexivity, symmetry, transitivity).

We write such a program directly into a suitable extension of the language L_m as follows:

Definition 6.2. Let L_{Syn} be the language obtained by adding to L_m the predicate name *synonymous*. Then a *definition of a synonymy* is a program *Syn* containing a set of facts like *synonymous*(r, r', λ) where r and r' are predicate names with the same arity and $\lambda \neq 0$, together with the rules

synonymous(R, R, 1)

synonymous(R', R, V_1)$\leftarrow$ *synonymous*(R, R', V_1)

synonymous(R, R'', V) $\leftarrow$ *synonymous*(R, R', V_1)$\wedge$ *synonymous*(R', R'', V_2)$\wedge$

$\wedge$ ($V = V_1 \otimes V_2$)

synonymous(R', R, 0).

Since we assume that in the language L there is only a finite set of predicate names, *Syn* is a finite set. We denote by *syn* the interpretation of *synonymous* in the least Herbrand model of *Syn*, namely

$$syn(r,r') = Sup\{\lambda \in \boldsymbol{B} : synonymous(r,r',\lambda) \in M_{Syn}\}. \qquad (6.1)$$

Due to the finiteness of *Syn*, the values of *syn* are in $\boldsymbol{B}$.

Definition 6.3. Let p be a fuzzy program and *Syn* be a definition of a synonymy. Then the *translation of p given Syn* is the classical program $\tau(p,Syn)$ in the language L_{Syn} obtained by adding to the program $\tau(p)$ the program *Syn* and the *synonymy rules*

$$th(R,X_1,...,X_n,V) \leftarrow synonymous(R, R', V_1) \wedge th(R', X_1,..., X_n, V_2) \wedge (V = V_1 \otimes V_2).$$

Notice that such a rule is strictly related with a rule considered in [25] (see the proof of Theorem 23). As it is usual, we denote by $M_{\tau(p,Syn)}$ and $T_{\tau(p,Syn)}$ the least Herbrand model and the one-step consequence operator of $\tau(p,Syn)$, respectively.

We are now ready to define a suitable notion of least Herbrand model for a similarity-based logic programming.

Definition 6.4. Let p be a fuzzy program and *Syn* the definition of a synonymy. Then we call *least Herbrand syn-model of p* the fuzzy set of facts $m_p^{Syn} : B_L \rightarrow V$ defined by setting, for every $r(t_1,...,t_n) \in B_L$,

$$m_p^{Syn}(r(t_1,...,t_n)) = Sup\{\lambda \in \boldsymbol{B} : th(r,t_1,...,t_n,\lambda) \in M_{\tau(p,Syn)}\}. \qquad (6.2)$$

7. A justification for the proposed synonymy logic

Obviously, the question arises whether Definition 6.4 is an adequate one for a synonymy logic. Now, usually one proves the adequateness of a logic by the exhibition of a completeness theorem. Unfortunately it is not evident if this can be done for synonymy logic since it is not evident whether the notion of synonymy is semantic in nature or not. Nevertheless it is possible to give some arguments in favour of such a definition. To do this, at first we prove the following useful proposition in which the synonymy *syn* is extended to B_L by setting

$syn(\alpha, \alpha') = syn(r,r')$ if $\alpha = r(t_1,...,t_n)$ and $\alpha' = r'(t_1,...,t_n)$

$syn(\alpha, \alpha') = 0$ otherwise.

Theorem 7.1. Assume that V is totally ordered and let α and α' be two facts. Then

$$m_p^{Syn}(\alpha')\otimes syn(\alpha,\alpha') \leq m_p^{Syn}(\alpha) \qquad (7.1)$$

and therefore,

$$syn(\alpha,\alpha') \leq (m_p^{Syn}(\alpha) \leftrightarrow m_p^{Syn}(\alpha')) \qquad (7.2)$$

Proof. In the case $syn(\alpha,\alpha') = 0$, then (7.1) is trivial. Otherwise, assume that $\alpha = r(t_1,...,t_n)$, $\alpha' = r'(t_1,...,t_n)$ and set

$A = \{\lambda \in \boldsymbol{B} : th(r,t_1,...,t_n,\lambda) \in M_{\tau(p,Syn)}\}$ and $A' = \{\lambda' \in \boldsymbol{B} : th(r',t_1,...,t_n,\lambda') \in M_{\tau(p,Syn)}\}$.

Then, by the synonymy rule for every $x \in \boldsymbol{B}$ such that $synonymous(r,r',x) \in M_{Syn}$ we have

$\lambda' \in A' \Rightarrow \lambda'\otimes x \in A$

and therefore, since in (6.1) the value $syn(r,r') = syn(\alpha,\alpha')$ is obtained as a maximum,

$\lambda' \in A' \Rightarrow \lambda'\otimes syn(\alpha,\alpha') \in A.$

So, $\{\lambda'\otimes syn(\alpha,\alpha') : \lambda' \in A'\} \subseteq A$ and, by the continuity of $\otimes$,

$m_p^{Syn}(r'(t_1,...,t_n))\otimes syn(\alpha,\alpha') = (Sup\{\lambda' \in A'\})\otimes syn(\alpha,\alpha') =$

$= Sup\{\lambda'\otimes syn(\alpha,\alpha'): \lambda' \in A'\} \leq Sup\{\lambda : \lambda \in A\} = m_p^{Syn}(r(t_1,...,t_n))$.

To prove (7.2) observe that (7.1) entails

$$syn(\alpha,\alpha') \leq m_p^{Syn}(\alpha') \rightarrow m_p^{Syn}(\alpha)$$

and therefore by symmetry

$$syn(\alpha,\alpha') \leq m_p^{Syn}(\alpha) \rightarrow m_p^{Syn}(\alpha').$$

Since in a totally ordered residuated lattice

$m_p^{Syn}(\alpha') \leftrightarrow m_p^{Syn}(\alpha) = (m_p^{Syn}(\alpha') \rightarrow m_p^{Syn}(\alpha)) \wedge (m_p^{Syn}(\alpha) \rightarrow m_p^{Syn}(\alpha'))$, and (7.2) follows.

Inequality (7.2) says that m_p^{Syn} is continuous with respect to *syn*, in a sense. We can express such a claim in a more precise way by referring to the following results proved by Valverde in [30].

Proposition 7.2. Let $h : [0,1] \to [0,\infty]$ be an additive generator, i.e. a strictly decreasing continuous map $h : [0,1] \to [0,\infty]$ such that $h(1) = 0$. Define the operation ⊗ by setting

$x \otimes y = h^{-1}(h(x)+h(y))$ if $h(x)+h(y) \leq h(0)$

$x \otimes y = 0$ otherwise.

Then ⊗ is an Archimedean triangular norm. If ↔ is the associated equivalence, then

$$h(x \leftrightarrow y) = |h(y) - h(x)|. \qquad (7.3)$$

As an example, if $h(x) = 1-x$, then ⊗ is the Łukasiewicz norm and $h(x \leftrightarrow y) = |y - x|$. In the case $h(x) = -log(x)$ for $x \neq 0$ and $h(0) = \infty$, ⊗ is the usual product and $h(x \leftrightarrow y) = |log(y)-log(x)|$.

Proposition 7.3. Assume that ⊗ is an Archimedean norm whose additive generator is the map $h : [0,1] \to [0,\infty]$, and let *syn* be a ⊗-synonymy. Then the map $d : B_L \times B_L \to [0,\infty]$ defined by setting

$d_{syn}(\alpha,\alpha') = h(syn(\alpha,\alpha'))$

for every α and α' in B_L is an extended pseudo-distance.

Theorem 7.4. In the case ⊗ is an Archimedean norm, the function $m_p^{Syn} : B_L \to [0,1]$ is a continuous map from the extended pseudo-metric space (B_L, d_{syn}) to [0,1].

Proof. By (7.2) and (7.3) for every pair α and α' of facts

$$d_{syn}(\alpha,\alpha') = h(syn(\alpha,\alpha')) \geq h(m_p^{Syn}(\alpha) \leftrightarrow m_p^{Syn}(\alpha')) = |h(m_p^{syn}(\alpha)) - h(m_p^{Syn}(\alpha'))|$$

This inequality entails that $h \circ m_p^{syn}$ is continuous. Since h is a injective continuous map defined in the compact set [0,1] and with values in a Hausdorff space, h^{-1} is continuous. So, we can conclude that m_p^{syn} is continuous, too.

8. Another justification.

Another argument in favour of Definition 6.4 is that it is in accordance with the abstract definition of a similarity logic given in [17]. In such a book an *abstract fuzzy logic* is defined as a continuous (conservative) operator $H : V^F \rightarrow V^F$ defined in the class V^F of all the fuzzy subsets of a given set F. H is named *the one-step consequence operator*. $H(s)$ is interpreted as the fuzzy subset of formulas we can obtain from s by an one-step proof. The *deduction operator* is the closure operator $D : V^F \rightarrow V^F$ generated by H. This means that, for every fuzzy subset s of formulas

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

Let *sim* be a synonymy relation, then a continuous operator $SYN : V^F \rightarrow V^F$ is defined by setting,

$$SYN(s)(\alpha) = Sup\{syn(\alpha',\alpha)\otimes s(\alpha') : \alpha'\in B_L\} \qquad (8.1)$$

$SYN(s)$ is interpreted as the fuzzy subsets of facts which are a synonymous of a fact in s. A fuzzy subset s of facts is a fixed point for SYN if and only if, for every α, $\alpha'\in B_L$

$$syn(\alpha',\alpha)\otimes s(\alpha')\leq s(\alpha). \qquad (8.2)$$

Definition 8.1. Let H be the one-step consequence operator of an abstract fuzzy logic and let SYN be a synonymy operator. Then we call *abstract synonymy logic* the abstract logic whose one-step consequence operator is the composition $H\circ SYN$ (see [17]).

In the case F coincides with B_L and H is the one-step consequence operator T_p of a fuzzy program p, we obtain an *abstract synonymy logic programming*. The Herbrand models of such a logic are the fixed points of $T_p\circ SYN$, i.e. the Herbrand models of T_p which are fixed points for SYN.

Theorem 8.2. The least Herbrand model m_p^{Syn} given in Definition 6.4 coincides with the least Herbrand model of the abstract synonymy logic defined by T_p and SYN.

Proof. We have to prove that m_p^{Syn} is a fixed point for both SYN and T_p and that if m is a fixed point for SYN and T_p then $m \supseteq m_p^{Syn}$. Now, from (7.1) it follows that m_p^{Syn} is a fixed point of SYN. To prove that m_p^{Syn} is a fixed point of T_p we have to prove that, for every fact α,

$$T^*_p(m_p^{Syn})(\alpha)\vee Fact(p)(\alpha) \leq m_p^{Syn}(\alpha).$$

Since $Fact(p)(\alpha)\leq m_p^{Syn}(\alpha)$, this is equivalent to prove that

$$T^*_p(m_p^{Syn})(\alpha) \leq m_p^{Syn}(\alpha)$$

and therefore that, given any ground rule $\alpha \leftarrow r_1(t^1_1,...,t^1_{n(1)})\wedge...\wedge r_m(t^m_1,...,t^m_{n(m)})$,

$$\mu\otimes m_p^{Syn}(r_1(t^1_1,...,t^1_{n(1)}))\otimes...\otimes m_p^{Syn}(r_m(t^m_1,...,t^m_{n(m)})) \leq m_p^{Syn}(\alpha)$$

where $\mu = Gr(p)(\alpha \leftarrow r_1(t^1{}_1,...,t^1{}_{n(1)})\wedge...\wedge r_m(t^m{}_1,...,t^m{}_{n(m)}))$. Now

$\mu\otimes m_p{}^{Syn}(r_1(t^1{}_1,...,t^1{}_{n(1)}))\otimes...\otimes m_p{}^{Syn}(r_m(t^m{}_1,...,t^m{}_{n(m)}))$ =

$= \mu\otimes(Sup\{\lambda_1 : th(r_1,t^1{}_1,...,t^1{}_{n(1)},\lambda_1)\in M_{\tau(p,Syn)}\}) \otimes...\otimes (Sup\{\lambda_m : th(r_m,t^m{}_1,...,t^m{}_{n(m)},\lambda_m) \in M_{\tau(p,Syn)}\}) =$

$= Sup\{\lambda_1\otimes\ldots\otimes\lambda_m\otimes\mu : th(r_1,t^1{}_1,...,t^1{}_{n(1)},\lambda_1) \in M_{\tau(p,Syn)}, ..., th(r_m,t^m{}_1,...,t^m{}_{n(m)},\lambda_m) \in M_{\tau(p,Syn)} \}$.

On the other hand, if α is the formula $r(t_1,...,t_n)$, the translation of the rule

$\alpha \leftarrow r_1(t^1{}_1,...,t^1{}_{n(1)})\wedge...\wedge r_m(t^m{}_1,...,t^m{}_{n(m)})$ is

$th(r, t_1,...,t_n,Z_{m+1}) \leftarrow th(r_1,t^1{}_1,...,t^1{}_{n(1)},Z_1)\wedge...\wedge th(r_m,t^m{}_1,...,t^m{}_{n(m)},Z_m) \wedge$

$\wedge (Z_{m+1} = Z_1\otimes...\otimes Z_m\otimes\mu)$.

Such a rule enables us to claim that if $th(r_1,t^1{}_1,...,t^1{}_{n(1)},\lambda_1) \in M_{\tau(p,Syn)},..., th(r_m,t^m{}_1,...,t^m{}_{n(m)},\lambda_m)\in M_{\tau(p,Syn)}$, then $th(r, t_1,...,t_n,\lambda_1\otimes...\otimes\lambda_m\otimes\mu)\in M_{\tau(p,Syn)}$. In turn, this entails that

$Sup\{\lambda_1\otimes\ldots\otimes\lambda_m\otimes\mu : th(r_1,t^1{}_1,...,t^1{}_{n(1)},\lambda_1) \in M_{\tau(p,Syn)}, ...,$

$th(r_m,t^m{}_1,...,t^m{}_{n(m)},\lambda_m) \in M_{\tau(p,Syn)}\} \leq Sup\{\lambda : th(r,t_1,...,t_n,\lambda) \in M_{\tau(p,Syn)}\} = m_p{}^{Syn}(r(t_1,...,t_n))$

Thus, $m_p{}^{Syn}$ is a fixed point of T_p.

Let m be a fixed point for both SYN and T_p. Then to prove that $m \supseteq m_p{}^{Syn}$, it is sufficient to prove that, given $\lambda\in \boldsymbol{B}$,

$th(r,t_1,...,t_n,\lambda) \in T_{\tau(p,Syn)}{}^n(\emptyset) \Rightarrow m(r(t_1,...,t_n)) \geq \lambda$

for every $n\in N$. We will prove this by induction on n. Indeed, in the case $n = 1$, since $T^*{}_p(m)\cup Fact(p) \subseteq m$, we have

$th(r,t_1,...,t_n,\lambda) \in T_{\tau(p,Syn)}(\emptyset) \Rightarrow th(r,t_1,...,t_n,\lambda) \in Fact(\tau(p))$

$\Rightarrow p(r(t_1,...,t_n)) = \lambda \Rightarrow m(r(t_1,...,t_n)) \geq \lambda.$

Assume that the implication holds true for n and that

$th(r,t_1,...,t_n,\lambda) \in T_{\tau(p,Syn)}{}^{n+1}(\emptyset) = T_{\tau(p,Syn)}(T_{\tau(p,Syn)}{}^n(\emptyset))$.

Then it is possible that $th(r,t_1,...,t_n,\lambda)$ is obtained by the rule

$th(r, t_1,...,t_n,\lambda) \leftarrow th(r_1,t^1{}_1,...,t^1{}_{n(1)},\lambda_1) \wedge...\wedge th(r_q,t^q{}_1,...,t^q{}_{n(q)},\lambda_q)\wedge(\lambda = \lambda_1\otimes...\otimes\lambda_q\otimes\mu)$

in $\tau(p)$ with $th(r_i,t^i{}_1,...,t^i{}_{n(i)},\lambda_i)\in T_{\tau(p,Syn)}{}^n(\emptyset)$. In such a case, by induction hypothesis, we have that $m(r_i(t^i{}_1,...,t^i{}_{n(i)})) \geq \lambda_i$. Since m is a fixed point for T_p,

$\lambda = \lambda_1\otimes...\otimes\lambda_m\otimes\mu \leq \mu\otimes m(r_1(t^1{}_1,...,t^1{}_{n(1)}))\otimes...\otimes m(r_q(t^q{}_1,...,t^q{}_{n(q)})) \leq m(r(t_1,...,t_n))$.

Assume that $th(r,t_1,...,t_n,\lambda)$ is obtained by the rule

$th(r,t_1,...,t_n,\lambda) \leftarrow synonymous(r, r',\lambda_1) \wedge th(r', t_1,...,t_n,\lambda_2) \wedge (\lambda = \lambda_1\otimes\lambda_2)$.

in Syn with $th(r', t_1,...,t_n,\lambda_2)\in T_{\tau(p,Syn)}{}^n(\emptyset)$. Then by inductive hypothesis $m(r'(t_1,...,t_n)) \geq \lambda_2$. Since m is a fixed point for SYN,

$\lambda = \lambda_1 \otimes \lambda_2 \leq \lambda_1 \otimes m(r'(t_1,...,t_n)) \leq m(r(t_1,...,t_n))$.

Finally, it is possible that $th(r,t_1,...,t_n,\lambda) \in Fact(\tau(p,Syn))$, i.e. that $th(r,t_1,...,t_n,\lambda) \in Fact(\tau(p))$. In such case we proceed as in the case $n = 1$.

Since both the operators T_p and SYN are continuous, in accordance with the fixed-point theorem for continuous operators (see for example [17]), we can obtain m_p^{syn} as the limit of the sequence

$$T_p(\emptyset) \subseteq SYN(T_p(\emptyset)) \subseteq T_p(SYN(T_p(\emptyset))) \subseteq . . .$$

Equivalently, since the operators $SYN \circ T_p$, $T_p \circ SYN$ and $T_p \vee SYN$ define the same class of fixed points, we can obtain m_p^{Syn} also as the limit of the sequence

$$SYN(\emptyset) \subseteq T_p(SYN(\emptyset)) \subseteq SYN(T_p(SYN(\emptyset))) \subseteq . . .$$

or of the sequence

$$(T_p \vee SYN)(\emptyset) \subseteq (T_p \vee SYN)((T_p \vee SYN)(\emptyset)) \subseteq (T_p \vee SYN)((T_p \vee SYN)((T_p \vee SYN)(\emptyset))) \subseteq ...$$

9. Recursive enumerability in fuzzy logic programming and in synonymy logic programming

In this section we analyze the computational features of the proposed logic by referring to the notion of recursively enumerable fuzzy subset.

In classical logic programming there is no difficulty to represent all the recursive enumerable subsets and this shows that the associated paradigm of computability is in accordance with Church thesis. We can formulate an analogous question for fuzzy logic programming and sinonymy-based logic programming. In order to do this we will refer to a notion of recursive enumerability for fuzzy subsets which is in accordance with the theory proposed in [2] and [19] .

Definition 9.1. We say that a continuous residuated lattice $(V, \leq, \otimes, \rightarrow, 0, 1)$ is *effective* provided that there is coding of its basis $\boldsymbol{B}$ such that

- the lattice operations and $\otimes$ are effectively computable in $\boldsymbol{B}$
- the relation $\prec$ is recursively enumerable in $\boldsymbol{B}$.

All the examples of continuous residuated lattices in Section 5 are also effective.

Definition 9.2. Let S be a coded set and $s : S \to V$ a fuzzy subset of S. Then we say that s is *recursively enumerable* provided that a computable function $h : S \times N \to \boldsymbol{B}$ exists which is increasing with respect to the second variable and such that, for every $x \in S$,

$$s(x) = Sup_{n\in N}\, h(x,n). \qquad (9.1)$$

Assume that in $(V, \leq, \otimes, \to, 0, 1)$ an involution ~ is defined which is computable in $\boldsymbol{B}$, then we can define the notion of *complement* $-s$ of a fuzzy subset s by setting $(-s)(x) = \sim s(x)$. Then we say that a fuzzy subset s is *recursively co-enumerable* if its complement $-s$ is recursively enumerable. If s is both recursively enumerable and recursively co-enumerable, the we say that s is *decidable*.

Definition 9.3. Let L be a first order language and U_L the related Herbrand universe. Then we say that a fuzzy subset $s : U_L \to V$ of U_L is *representable by a fuzzy program p* provided that a predicate name r exists such that $s(x) = m_p(r(x))$ for every x in U_L.

Theorem 9.4. Consider a finite fuzzy program p with truth values in $\boldsymbol{B}$. Then, m_p is recursively enumerable. Consequently every fuzzy subset of U_L representable by a fuzzy program is recursively enumerable.

Proof. Since M_p is recursively enumerable, we can define the function $h : B_L \times N \to \boldsymbol{B}$ as follows. Let $r(t_1,\dots,t_k)$ be an element in B_L, then

- we generate step-by-step all the elements $\lambda_1,...,\lambda_i,...$ of the set $\{\lambda \in \boldsymbol{B} : th(r,t_1,\dots,t_k,\lambda) \in M_p\}$
- at the same time, we generate the increasing sequence $(h(r(t_1,\dots,t_k),n))_{n\in N}$ by setting $h(r(t_1,\dots,t_k),1) = \lambda_1$, $h(r(t_1,\dots,t_k),i) = h(r(t_1,\dots,t_k),i\text{-}1) \vee \lambda_i$

It is evident that h is computable and order-preserving with respect to the second variable and that

$$m_p(r(t_1,\dots,t_k)) = Sup_{n\in N}\, h(r(t_1,\dots,t_k), n). \qquad (9.2)$$

Now, the question arises whether every recursively enumerable fuzzy subset can be represented in such a way or not. Unfortunately, the answer is negative.

Theorem 9.5. There are recursively enumerable fuzzy subsets which are not definable by a fuzzy program.

Proof. In [2] one defines *d-enumerable* a recursively enumerable fuzzy subset in which instead of (9.1) we have that $s(x) = Max_{n\in N}\, h(x,n)$, i.e. every $s(x)$ is obtained as a maximum of the sequence $h(x,n)$. Also, one proves that, in the case V is the interval [0,1], there is a recursively enumerable fuzzy subset s which is not *d*-recursively enumerable. On the other hand, since by Lemma 5.5 the sequence $(h(r(t_1,\dots,t_k),n))_{n\in N}$ admits a maximum, all the fuzzy subsets representable by a fuzzy program are *d*-enumerable.

It is evident that we can extend Theorems 9.4 and 9.5 to the synonymy-based logic programming.

We conclude by observing that (apparently) we can obtain $m_p^{syn}(r(t_1,\dots,t_n))$ by the *findall* operation in *Prolog* and by a predicate enabling us to calculate the maximum of a list. Indeed, we can consider the rule

$Herbrand\_model(r,t_1,...,t_n,Z) \leftarrow findall(Z_1,th(r,t_1,...,t_n,Z_1), List) \wedge maximum(List,Z)$

or, in a most general way:

$Herbrand\_model(R,X_1,...,X_n,Z)) \leftarrow findall(Z_1,th(R,X_1,...,X_n,Z_1),List) \wedge maximum(List,Z)$

Regrettably, in spite of $\{\lambda \in V : th(r,t_1,...,t_n,\lambda)\}$ is finite, there is no general criterion to establish if all the elements in such a set where attained at a given step of the computation. Equivalently, in spite of the fact that we can compute the increasing sequence $h(r(t_1,\dots,t_k),1) \leq h(r(t_1,\dots,t_k),1) \leq \dots$ and that such a sequence becomes constant after a finite number of steps, there is no general criterion to establish if the maximum is attained at a given step. Obviously, this is not surprising since it is in accordance with the notion of recursive enumerability.

CHAPTER 4

KRIPKE-BASED BILATTICE LOGIC

1. Bilattice and fuzzy logic

Bilattice theory was introduced by Ginsberg [20] in order to treat both truth and grade of information from an algebraic point of view (see also Fitting [10]); its principal task is to give successful tools for logic programming. Formal fuzzy logic (or fuzzy logic in narrow sense) is a chapter of formal logic strictly related with the theory of fuzzy subsets and connected with the tradition of multi-valued logic (see [17], [21], [22], [27], [28], [34]).

Our aim is to investigate the potentialities of bilattice theory for the graded approach to formal fuzzy logic; in particular we show that bilattice theory enables us to obtain in a sense, a nice extension of the fuzzy logic.

Notice that in the literature about fuzzy logic an analogous of the notion of bilattice is considered under the name of intuitionistic fuzzy logic (see for example [8], [26]), but in the intutionistic approach there are some other limitations.

Our approach is different since we refer to a formal definition of fuzzy logic in Pavelka's sense in which a deduction apparatus is defined by a suitable fuzzy subset of logical axioms and by fuzzy inference rules. So, we propose and discuss some possible general definitions involving bilattice theory and extending Pavelka's ideas [28]. Also, to give an example, we apply the proposed apparatus to a Kripke-like logic related with a logic proposed by Ginsberg in its basic paper [20].

The main tool we use in this chapter is the notion of closure operator and the associated one of closure system. This in accordance with the abstract approach to fuzzy logic proposed in [17] in which Tarski's ideas of a logic as a closure operator is embraced. Recall that, given a complete lattice L, a *closure operator* in L is a map

$H : L \rightarrow L$ such that

$$H(x) \geq x \ ; \ x \geq y \Rightarrow H(x) \geq H(y) \ ; \ H(H(x)) = H(x).$$

In particular, if two closure operators have the same fixed points, they coincide. Finally, observe that, given $M \subseteq L$, *the closure system generated by M*, that we denote by $<M>$, is the intersection of all the closure systems containing M. Equivalently, $<M> = \{inf_{i\in I}\, m_i : (m_i)_{i\in I}$ is a family of elements of $M\}$. Also, $<M>$ coincides with the set of fixed points of H_M.

2. Bilattice theory

A bilattice is a structure with two lattice orderings: one ordering $\leq_t$ is with respect to the degree of truth, the other ordering $\leq_k$ is related with information or knowledge.

Definition 2.1. A *bilattice* is a structure $\boldsymbol{B} = (B, \leq_t, \leq_k, False, True, \bot, \mathrm{T})$ such that $(B, \leq_t, False, True)$ and $(B, \leq_k, \bot, \mathrm{T})$ are bounded lattices. If both the orders are complete, then we say that $\boldsymbol{B}$ is *complete*. We denote by $\wedge_t$ and $\vee_t$, $\wedge_k$, and $\vee_k$ the lattice operations in $(B, \leq_t, False, True)$ and in $(B, \leq_k, \bot, \mathrm{T})$, respectively; $\boldsymbol{B}$ is *interlaced* if all these operations are order preserving with respect to $\leq_t$ and $\leq_k$; $\boldsymbol{B}$ is *distributive* if all 12 distributive laws connecting $\wedge_t$, $\vee_t$, $\wedge_k$, and $\vee_k$ are valid; $\boldsymbol{B}$ satisfies *the decomposition property* provided that, for every $x\in B$,

$$x = (x\wedge_k True)\vee_k(x\wedge_k False).$$

It is easy to prove that if a bilattice is distributive, then it is also interlaced and that an interlaced bilattice satisfies the decomposition property.

Definition 2.2. Assume that in a bilattice $\boldsymbol{B}$ an operation $\sim : B\to B$ is defined in such a way that:

1. $x \leq_t y \Rightarrow {\sim}y \leq_t {\sim}x$
2. $x \leq_k y \Rightarrow {\sim}x \leq_k {\sim}y$
3. ${\sim\sim}x = x$.

Then we say that $(B, \leq_t, \leq_k, \sim, False, True, \bot, \mathrm{T})$ is a *bilattice with negation*.

Observe that since ~ is order-reversing with respect to $\leq_t$ and order-preserving with respect to $\leq_k$,

$${\sim}(x\wedge_t y) = {\sim}(x)\vee_t{\sim}(y) \ ; \ {\sim}(x\vee_t y) = {\sim}(x)\wedge_t{\sim}(y) \ ;$$

$${\sim}(x\wedge_k y) = {\sim}(x)\wedge_k{\sim}(y) \ ; \ {\sim}(x\vee_k y) = {\sim}(x)\vee_k{\sim}(y)$$

for every x, y in B. It is also immediate that ${\sim}False = True$, ${\sim}True = False$, ${\sim}\bot = \mathrm{T}$, ${\sim}\mathrm{T} = \bot$.

There are several ways to define a bilattice by starting from a bounded lattice $\boldsymbol{L} = (L, \leq, 0, 1)$. A way is to consider the set of intervals of L (see for example [29]) and it is related in a natural way with multi-valued logic. Indeed, an interval is interpreted as a constraint on a possible truth value.

Theorem 2.3. Let $I(L)$ be the set of closed intervals of a bounded lattice L (included the empty set) and define the structure $\boldsymbol{I(L)} = (I(L), \leq_t, \leq_k, \{0\}, \{1\}, [0,1], \emptyset)$ in such a way that

- $\leq_k$ is the dual of the inclusion relation,
- for every $[a,b]$, $[c,d]$ in $I(L)$-$\{\emptyset\}$, $[a,b] \leq_t [c,d]$ provided that $a\leq c$ and $b\leq d$,
- $\{0\} \leq_t \emptyset \leq_t \{1\}$ and $\emptyset$ is not t-comparable with any other interval.

Then $\boldsymbol{I}(\boldsymbol{L})$ is a bilattice which satisfies the decomposition property. If L is complete, then $\boldsymbol{I}(\boldsymbol{L})$ is complete. Moreover, if $-$ is an involution in L, by setting

$\sim[a,b] = [-b,-a]$; $\sim\varnothing = \varnothing$

we obtain a negation in $I(L)$. If L is different from the Boolean algebra $\{0,1\}$, $\boldsymbol{I}(\boldsymbol{L})$ is not interlaced.

Proof. We observe only that, for every interval $[a,b]$,

$[a,b] = [a,1]\cap[0,b] = ([a,b]\wedge_k\{1\})\vee_k([a,b]\wedge_k\{0\})$

and that,

$\varnothing = \{1\}\vee_k\{0\} = (\varnothing\wedge_k\{1\})\vee_k(\varnothing\wedge_k\{0\})$.

Moreover, due to the behavior of $\varnothing$, in the case $L \neq \{0,1\}$, $\boldsymbol{I}(\boldsymbol{L})$ is not interlaced. Indeed, if c is an element of L different from 0 and 1, then $[0,0] \leq_t [c,c]$ while $[0,0]\vee_k[c,1] = \varnothing$ and $[c,c]\vee_k[c,1] = [c,c]$. On the other hand the relation $\varnothing\leq_t[c,c]$ is false.

Observe that the lattice operations in $\boldsymbol{I}(\boldsymbol{L})$ are defined by setting

- $[a,b]\wedge_t[c,d] = [a\wedge c, b\wedge d]$; $[a,b]\vee_t[c,d] = [a\vee c, b\vee d]$
- $\{1\}\wedge_t\varnothing = \varnothing\wedge_t\{1\} = \varnothing$; $\{0\}\vee_t\varnothing = \varnothing\vee_t\{0\} = \varnothing$
- $\{1\}\vee_t\varnothing = \varnothing\vee_t\{1\} = \{1\}$; $\{0\}\wedge_t\varnothing = \varnothing\wedge_t\{0\} = \{0\}$
- $[a,b]\wedge_t\varnothing = \varnothing\wedge_t[a,b] = \{0\}$ ($[a,b]\neq\{1\}$) ;
  $[a,b]\vee_t\varnothing = \varnothing\vee_t[a,b] = \{1\}$ ($[a,b] \neq \{0\}$)
- $\varnothing\wedge_t\varnothing = \varnothing\vee_t\varnothing = \varnothing$.
- $[a,b]\wedge_k[c,d] = [a\wedge c, b\vee d]$; $[a,b]\vee_k[c,d] = [a\vee c, b\wedge d]$
- $[a,b]\wedge_k\varnothing = \varnothing\wedge_k[a,b] = [a,b]$; $[a,b]\vee_k\varnothing = \varnothing\vee_k[a,b] = \varnothing$
- $\varnothing\wedge_k\varnothing = \varnothing\vee_k\varnothing = \varnothing$.

Definition 2.4. Given a bounded lattice L (with an involution $-$), the bilattice $\boldsymbol{I}(\boldsymbol{L})$ is called the *interval bilattice* (*with negation*) *associated with L.*

Another very famous way to obtain a bilattice is the following one.

Theorem 2.5. Let $\boldsymbol{L} = (L,\leq,0,1)$ be a bounded lattice and denote by $\boldsymbol{B}(\boldsymbol{L})$ the structure $(L\times L, \leq_t, \leq_k, \sim, (0,1), (1,0), (0,0), (1,1))$ where $\sim$ is defined by setting $\sim(x,x') = (x',x)$, and the relations $\leq_t$, $\leq_k$ are defined by setting

$(x,x') \leq_t (y,y') \Leftrightarrow x \leq y$ and $x' \geq y$ and

$(x,x') \leq_k (y,y') \Leftrightarrow x \leq y$ and $x' \leq y$.

Then $\boldsymbol{B}(\boldsymbol{L})$ is an interlaced bilattice with negation. If L is complete (distributive) then $\boldsymbol{B}(\boldsymbol{L})$ is complete (distributive, respectively).

Definition 2.6. We call the *product bilattice associated with* ***L*** the bilattice ***B*****(*****L*****).**

Since ***B*****(*****L*****)** is interlaced, it satisfies the decomposition property. On the other hand, since $(x,x')\wedge_k(1,0) = (x,0)$ and $(x,x')\wedge_k(0,1) = (0,x')$, trivially

$$x = ((x,x')\wedge_k(1,0))\vee_k((x,x')\wedge_k(0,1)).$$

The following proposition shows a connection between the bilattices ***I*****(*****L*****)** and ***B*****(*****L*****)**.

Proposition 2.7. Let ***L*** be a bounded lattice with an involution $-$ and let $I_0(L)$ be the set of nonempty intervals of ***L***. Then by setting $h([a,b]) = (a, -b)$ we obtain an embedding of the structure $\boldsymbol{I_0(L)} = (I_0(L), \leq_t, \leq_k, \sim, \{0\}, \{1\}, [0,1])$ into the structure $(L\times L, \leq_t, \leq_k, \sim, (0,1), (1,0), (0,0))$.

3. Bilattice-based fuzzy logic: the semantics

In this section we call *valuation structure* a complete lattice $\boldsymbol{V} = (V, \leq, 0, 1)$ with $0\neq 1$. The elements in ***V*** are interpreted as truth values and, in particular, the minimum 0 and the maximum 1 are interpreted as “false” and true”, respectively.

To connect bilattice theory with fuzzy logic we interpret the elements in a bilattice ***B*** as pieces of information on the elements in *V*. To do this, we need to define a relation from *V* to *B*. The following is a possible definition.

Definition 3.1. A *bt-system* is a structure $(\boldsymbol{V}, \boldsymbol{B}, \models^*)$ such that ***V*** is a valuation structure, ***B*** is a complete bilattice and $\models^* \subseteq V\times B$ is a relation such that,

i) $\lambda \models^* x$ and $x' \leq_k x \Rightarrow \lambda \models^* x'$

ii) for every $\lambda \in V$ the set $\{x\in B : \lambda \models^* x\}$ admits a *k*-maximum

iii) $0 \models^* \mathit{False}$; $1 \models^* \mathit{True}$.

In the case the relation $\lambda\models^* x$ is satisfied, we say that λ *satisfies* x or that x is *a correct piece of information on* λ.

Definition 3.2. Given a *bt*-system $(\boldsymbol{V}, \boldsymbol{B}, \models^*)$, we set $Sat = \{x\in B :$ there is $\lambda\in V$ such that $\lambda \models^* x\}$ and we define the map $i : V \to B$ by setting

$i(\lambda) = Max_k\{x\in B : \lambda \models^* x\}$.

Also, we put

$Maxsat = \{x \in B : x$ is maximal in $(Sat, \leq_k)\}$;

$Compl = \{x \in B :$ there is $c \in Maxsat$, $x \geq_k c\}$.

We say that *Sat* is the set of *satisfiable* elements of *B*. We say that $i : V \to B$ is the *information map* and this since $i(\lambda)$ summarizes the whole information on λ we can obtain in the *bt*-system $(\boldsymbol{V}, \boldsymbol{B}, \models^*)$. Obviously, for every $\lambda \in V$,

$\lambda \models^* x \Leftrightarrow x \leq_k i(\lambda)$

and, consequently,

$Sat = \{x \in B :$ there is $c \in Maxsat$ such that $x \leq_k c\}$.

The elements in *Maxsat* are the maximal elements in *Sat*, if $x \in Compl$ the we say that *x* is *complete*.

Usually, the semantics in a multi-valued logic is defined in a truth-functional way. This means that if, for example, we consider a propositional language whose logical connectives are $\vee$, $\wedge$, $\neg$, then suitable operations $\oplus$, $\otimes$, $-$ are defined in *V* to interpret these connectives. Denoting by F the set of formulas, the semantics is obtained by considering the class M of truth assignments $m : \mathrm{F} \to V$ such that

$m(\alpha \wedge \beta) = m(\alpha) \otimes m(\beta)$; $m(\alpha \vee \beta) = m(\alpha) \oplus m(\beta)$; $m(\neg \alpha) = -m(\alpha)$.

Nevertheless, since there are interesting semantics which are not truth-functional (see [17]) we prefer the following abstract definition of semantics proposed in [28].

Definition 3.3. Let $\boldsymbol{V}$ be a valuation structure and F be the set of formulas in a given logical language. Then a *semantics* is a class *M* of maps $m : \mathrm{F} \to V$. The elements in *M* are called *models.*

To proceed in our definitions, we refer to the expressive language of fuzzy logic. We denote by L^S the class of all the *L*-subsets of *S*. Such a class is a complete lattice, the direct power of $\boldsymbol{L}$ with index set *S*. The order relation in L^S is denoted by $\subseteq$ and named *inclusion relation*. The meet and the join in L^S are denoted by $\cap$ and $\cup$ and named *intersection* and *union*, respectively. Finally, in the case a *negation* $- : L \to L$ is defined in *L*, the *complement of s* is the *L*-subset $-s$ defined by setting $(-s)(x) = -s(x)$ for every $x \in S$. In such a paper we are mainly interested in considering the set F of formulas of a given language and the knowledge order in a bilattice $\boldsymbol{B}$.

Definition 3.4. Given a bilattice $\boldsymbol{B}$, we call *B-subset of formulas* or *valuation* any element $v : \mathrm{F} \to B$ in B^{F}. We denote by $\supseteq_k$ the knowledge order in B^{F} and we call it *k-inclusion*. Also, we denote by $v_\perp$ and v_T the minimum and the maximum with respect to $\subseteq_k$ and we say that $v_\perp$ is the *empty information* and that v_T is the *totally inconsistent information.* The lattice operations, we denote by $\cap_k$ and $\cup_k$, are called *k-intersection* and *k-union*, respectively. Finally, we say that *v* is *pointwise satisfiable* if $v(\alpha) \in Sat$ for every formula α. In the case a negation ~ is defined in $\boldsymbol{B}$, we say that *v* is *balanced* if $v(\neg\alpha) = \sim v(\alpha)$.

Definition 3.5. Let $M \subseteq V^{\mathrm{F}}$ be a semantics and v be a B-set of formulas. We say that $m \in M$ *is a model of* v, in brief $m \models v$, if $m(\alpha) \models^* v(\alpha)$ for every formula α. In such a case we say that v *is satisfiable*.

Obviously, if v is satisfiable then v is pointwise satisfiable, too.

Definition 3.6. Given $m \in \mathrm{M}$, we denote by $\underline{m} : \mathrm{F} \rightarrow B$ the composition $i \circ m$ and we set $\underline{M} = \{\underline{m} : m \in \mathrm{M}\}$. Also, we define the *logical consequence operator* $L_c : B^{\mathrm{F}} \rightarrow B^{\mathrm{F}}$ by setting as usual,

$$Lc(v)(\alpha) = Inf_k\{m(\alpha) : \ m \models v\} \qquad (3.1)$$

for every $v \in B^{\mathrm{F}}$ and $\alpha \in \mathrm{F}$.

An element $\underline{m} \in \underline{\mathrm{M}}$ represents the way we can represent a world m by our information system ($\boldsymbol{V}$, $\boldsymbol{B}$, $\models^*$). Also, for every formula α, $L_c(v)(\alpha)$ is the information on the truth value of α shared by all the possible models of v. Namely, such an information says that the unknown truth value of α belongs to $\{\lambda \in V : \lambda \models^* L_c(v)(\alpha)\}$.

The proof of the following proposition is trivial.

Proposition 3.7. For every valuation v and $m \in M$,

$$m \models v \Leftrightarrow \underline{m} \supseteq_k v$$

and therefore,

$$L_c(v) = \cap_k\{\underline{m} : \ \underline{m} \supseteq_k v\} \qquad (3.2)$$

We can identify $\underline{\mathrm{M}}$ with the class of complete theories and the B-set $L_c(v)$ of logical consequences of v with the k-intersection of all the complete theories containing v. It is also of some interest to define an analogous of the notion of set of tautologies.

Definition 3.8. Given a semantics M, we call *B-subset of tautologies* the B-subset of formulas

$$Tau = L_c(v_{\perp}) = \cap_k\{\underline{m} : m \in \mathrm{M}\} \qquad (3.3)$$

In other words, $Tau(\alpha)$ is the information on α shared by all the possible models in the given semantics. The information content of $Tau(\alpha)$ is logical in nature since it depends on the structure of α and not on the state of the affairs. The proof of the following theorem is trivial.

Theorem 3.9. L_c is a closure operator in the lattice $(B^{\mathrm{F}}, \subseteq_k)$. Namely, L_c is the closure operator generated by $\underline{M}$.

4. The deduction apparatus

We define a notion of deduction apparatus by extending the classical notions of inference rule and of set of logical axioms. The definitions are inspired to the ones given by Pavelka [28].

Definition 4.1. Let ***B*** be a complete bilattice, then an *n-ary B-inference rule* is a pair $r = (r_{sin}, r_{sem})$ where r_{sin} is a partial n-ary operation in F (i.e. an inference rule in the usual sense) and r_{sem} is an n-ary operation in *B* preserving the inductive limits, i.e. arbitrary k-joins of k-directed subsets of *B* (*continuity property*). A *B-deduction apparatus*, in brief a *deduction apparatus*, is a pair (*IR*, *la*) such that *la* is a *B*-subset of formulas, we call *B-subset of logical axioms*, and *IR* is a set of *B*-inference rules.

We represent an application of an n-ary *B*-inference rule as follows

$$\langle \frac{\alpha_1,\dots,\alpha_n}{r_{\sin}(\alpha_1,\dots,\alpha_n)} \mid \frac{\lambda_1,\dots,\lambda_n}{r_{sem}(\lambda_1,\dots,\lambda_n)} \rangle$$

The intended meaning is that if $\lambda_1,\dots,\lambda_n$ are correct piece of information on $\alpha_1,\dots,\alpha_n$, then $r_{sem}(\lambda_1,\dots,\lambda_n)$ is a correct piece of information on the formula $r_{sin}(\alpha_1,\dots,\alpha_n)$.
Every deduction apparatus is associated with a notion of proof in the following way.

Definition 4.2. A *proof* π *of a formula* α is any 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 have to indicate also the rule and the formulas $\alpha_{i(1)},\dots, \alpha_{i(n)}$ in $\alpha_1,..., \alpha_{i-1}$ used to obtain α_i).

Differently from the classical logic, the justifications are necessary to calculate the information furnished by a proof.
Observe that, as in the classical case, for any $i \leq m$, the initial segment $\alpha_1,..., \alpha_i$ of a proof $\alpha_1,..., \alpha_m$ is a proof of α_i we denote by $\pi(i)$.

Definition 4.3. Given a proof $\pi = \alpha_1,...,\alpha_m$ of α and a valuation $v : F\to B$, the *information on* α *furnished by* π *given v* is the element $I(\pi,v)$ in *B* defined by induction on the length of π in accordance with the following rules:

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

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

$I(\pi,v) = r_{sem}(I(\pi_{i(1)},v),\ldots,I(\pi_{i(n)},v))$ if α_m is obtained by a rule $r = (r_{sin}, r_{sem})$ from $\alpha_{i(1)},\ldots, \alpha_{i(n)}$ with $i(1) < m,\ldots, i(n) < m$.

Notice 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. So, the first two lines in the definition of $I(\pi,v)$ give also the induction basis.

Different proofs of the same formula α can give different pieces of information on the truth value of α. This suggests the following definition.

Definition 4.4. Given a deduction apparatus (*IR, la*), we call *deduction operator* the operator $D : B^F \to B^F$ defined by setting, for every $v \in B^F$ and $\alpha\in$ F,

$$D(v)(\alpha) = Sup_k\{I(\pi,v) : \pi \text{ is a proof of } \alpha\} \qquad (4.1)$$

If v is a fixed point for D, then we say that v is a *theory*.

It useful to assume that in the considered deduction apparatus there is the *fusion rule*:

$$\langle\frac{\alpha \quad \alpha}{\alpha} \mid \frac{x \quad y}{x\vee_k y}\rangle$$

Such a rule enables us to fuse two different proofs π_1 and π_2 of a formula α into an unique proof π of α in such a way that $I(\pi,v) = I(\pi_1,v)\vee_k I(\pi_2,v)$. This entails that the set $\{I(\pi,v) : \pi$ is a proof of $\alpha\}$ is closed with respect to $\vee_k$ and therefore is up-ward directed. Then, the value $D(v)(\alpha)$ is the direct limit of an up-ward directed class. On the other hand, if we add to a deduction system the fusion rule the power of the system remains unchanged since this rule gives no contribution to the definition of D.

The continuity property of the inference rules enables us to prove the following theorem.

Theorem 4.5. The deduction operator D is a closure operator in the lattice $(B^F, \subseteq_k)$.

Proof. To prove that $D(v)\supseteq_k v$ it is sufficient to observe that, given a formula α, the formula α justified as an hypothesis is a proof π of α such that $I(\pi,v) = v(\alpha)$. We can prove that D is monotone by proving that $I(\pi,v)$ is monotone with respect to v for every proof π of a formula α. To this aim it is sufficient to observe that the semantics part of the inference rules is monotone and to proceed by induction on the length of π. To prove that D is idempotent we have to prove that $D(v)$ is a fixed point for D and therefore that, given a formula α,

$$Sup_k\{I(\pi,D(v)) : \pi \text{ is a proof of } \alpha\} \leq_k D(v)(\alpha).$$

Equivalently, we have to prove that, for every proof $\pi = \alpha_1,\ldots,\alpha_m$ of α,

$$I(\pi,D(v)) \leq_k D(v)(\alpha) \qquad (4.2)$$

We will proceed by induction on the length m of π. Now, if α_m is assumed either as a logical axiom or as an hypothesis, then (4.2) is evident. Otherwise, assume that α_m is obtained by an n-ary inference rule and therefore that

$$I(\pi,D(v)) = r_{sem}(I(\pi_{i(1)}, D(v)),\dots,I(\pi_{i(n)}, D(v)))$$

where $\pi_{i(1)},\dots,\pi_{i(n)}$ are the proofs of the formulas $\alpha_{i(1)},\dots, \alpha_{i(n)}$, $i(1) < m,\dots, i(n)<m$. Then, taking in account the induction hypothesis, the definition of $D(v)$ and the continuity property of r_{sem},

$I(\pi,D(v)) \leq_k r_{sem}(D(v)(\alpha_{i(1)}),\dots,D(v)(\alpha_{i(n)}))$

$= r_{sem}(Sup_k\{I(\underline{\pi},v) : \underline{\pi}$ is a proof of $\alpha_{i(1)}\},\dots, Sup_k\{I(\underline{\pi},v) : \underline{\pi}$ is a proof of $\alpha_{i(n)}\})=$

$= Sup_k\{r_{sem}(I(\underline{\pi}_{i(1)},v),\dots, I(\underline{\pi}_{i(n)},v)) : \underline{\pi}_{i(1)}$ is a proof of $\alpha_{i(1)},\dots, \underline{\pi}_{i(n)}$ is a proof of $\alpha_{i(n)}\}) \leq_k D(v)(\alpha)$.

We are now ready to give the main definitions in this paper.

Definition 4.6. Let M be a fuzzy semantics and (IR, la) a deduction apparatus. Then (IR, la) is *correct* with respect to M if $L_c(v) \sqsupseteq_k D(v)$ for every $v \in B^F$. (IR, la) is *complete* with respect to M if $D(v) \sqsupseteq_k L_c(v)$ for every $v \in B^F$. In the case (IR, la) is both correct and complete, i.e. $D = L_c$, we say that (M, IR, la) is a *bilattice based fuzzy logic* and that *the completeness theorem* holds true.

5. Examples of *bt*-systems

In order to illustrate the notion of bilattice based fuzzy logic, we will give some example. The first one is related with the interval bilattices.

Proposition 5.1. Let V be a valuation structure and assume that B is the interval bilattice ***I*(*V*)**. Then we obtain a *bt*-system by setting

$$\lambda \models^* x \Leftrightarrow \lambda \in x.$$

In such a system,

$Sat = I(V)-\{\varnothing\}$; $i(\lambda) = \{\lambda\}$; $Maxsat = \{x\in I(V) : x$ is a singleton$\}$

$Compl = Maxsat \cup\{\varnothing\}$.

In accordance, given a semantics M,

$m \models v$ provided $\Leftrightarrow$ for every formula α, $m(\alpha)\in v(\alpha)$,

and, if v admits a model, $L_c(v)(\alpha)$ is the least interval containing $\{m(\alpha) : m \models v\}$, i.e.

$L_c(v)(\alpha) = [Inf\{m(\alpha) : m \models v\}, Sup\{m(\alpha) : m \models v\}]$.

In particular, for every formula α,

$Tau(\alpha) = [Inf\{m(\alpha) : m\in \mathrm{M}\}, Sup\{m(\alpha) : m \in \mathrm{M}\}]$.

As an example, assume that the valuation structure is ([0,1], $\wedge$, $\vee$, 1-x). Then $Tau(\alpha\vee\neg\alpha) = [0.5, 1]$ and $Tau(\alpha\wedge\neg\alpha) = [0, 0.5]$. This means that by our formalisms we can have useful a-priori information on the formulas. Instead, if we adopt the usual notions of tautology and contradictions no tautology or contradiction exists in such a logic. Notice also that while in classical logic we refer both to the notion of tautology and contradiction to represent the a-priori information of the formulas, in our approach $Tau(\alpha)$ represents the whole a-priori information we have on α.

The definition of a *bt*-system in the case of a product bilattice ***B*(*V*)** is more problematic. Assume that a negation $-$ in V exists. Then a definition of a *bt*-system have to be in accordance with the embedding h of $\boldsymbol{I_0(V)}$ into ***B*(*V*)** defined in Theorem 2.9. This means that we have to assume that for every (a,b) such that $a \leq -b$, $\lambda \models^* (a,b)$ if and only if $\lambda\in [a, -b]$. This suggests the following definition.

Proposition 5.2. Assume that a negation $-$ in V exists and that ***B*** is the product bilattice ***B*(*V*)**. Then we obtain a *bt*-system by setting

$\lambda \models^* (a,b) \;\Leftrightarrow\; a \leq \lambda$ and $b \leq -\lambda$.

In such a *bt*-system,

$Sat = \{(a,b) : a \leq -b\}$; $i(\lambda) = (\lambda,-\lambda)$;

$Maxsat = \{(a, b) : a = -b\}$;

$Compl = \{(\lambda,\mu)$: there is x such that $\lambda\geq x, \mu \geq -x\}$.

Also, given a semantics M,

$m \models v \;\Leftrightarrow\; m \geq v_+$ and $-m \geq v_-$

and,

$L_c(v)(\alpha) = (Inf\{m(\alpha): m \models v\}, Inf\{-m(\alpha) : m \models v\})$.

Proof. We observe only that

$\lambda \models^* (a,b) \;\Leftrightarrow\; a \leq \lambda$ and $b \leq -\lambda \;\Leftrightarrow\; a \leq \lambda \leq -b$

and therefore that there is λ such that $\lambda \models^* (a,b)$ if and only if $a \leq -b$. Moreover

$L_c(v)(\alpha) = Inf_k\{\underline{m}(\alpha) : m \models v\} = Inf_k\{(m(\alpha),-m(\alpha)) : m \models v\} =$

$= (Inf\{m(\alpha) : m \models v\}, Inf\{-m(\alpha) : m \models v\})$.

In particular,

$Tau(\alpha) = (Inf\{m(\alpha) : m\in \mathrm{M}\}, Inf\{-m(\alpha) : m\in \mathrm{M}\})$.

So, by referring to the valuation structure ([0,1],$\wedge$,$\vee$,1-x, 0, 1),

$Tau(\alpha\vee\neg\alpha) = (0.5,0)$ and $Tau(\alpha\wedge\neg\alpha) = (0, 0.5)$.

6. The proposed approach extends the one of Pavelka

In this section we will show that the definitions proposed in this paper extend the usual ones in the graded approach to fuzzy logic (see [28]). This in spite of the fact that in such an approach it is not apparent the reference to a knowledge order since one refers to the order relation ≤ in *V*. As a matter of fact the reference to the knowledge order is implicit in the proposed semantics since the information is represented by a fuzzy subset v: F $\rightarrow V$ of formulas and one claims that m is a model of v provided that $m \supseteq v$. This means that the information carried on by v is that, for every formula α, $v(\alpha)$ represents a lower-bound constraint like "the truth value of α is greater or equal to $v(\alpha)$". Then in the graded approach one manages interval constraints on truth values and not truth values and we have not confuse the truth value λ with the constraint $[\lambda,1]$. To make more precise such an observation, we consider the following sub-bilattice of ***I*(*V*)**.

Proposition 6.1. Let ***L*** be a bounded lattice and consider the set

$I^+(L) = \{[a,1] : a \in L\} \cup \{\{0\}, \varnothing\}$.

Then the substructure $\boldsymbol{I^+(L)}$ of $\boldsymbol{I(L)}$ defined by $I^+(L)$ is a bilattice satisfying the decomposition property, we call such a bilattice the *lower-bound bilattice associated with **L***.

Then in such a bilattice $\leq_k$ is the dual of the inclusion relation and $\leq_t$ is defined by setting

- $\{0\}$ is the minimum with respect to $\leq_t$
- $[a,1] \leq_t [c,1] \Leftrightarrow a \leq c$
- $\{0\} \leq_t \varnothing \leq_t \{1\}$ and $\varnothing$ is not t-comparable with any other interval.

In accordance, the operations are defined by setting

- $[a,1] \wedge_t [c,1] = [a \wedge c, 1]$; $[a,1] \vee_t [c,1] = [a \vee c, 1]$
- $\{1\} \wedge_t \varnothing = \varnothing \wedge_t \{1\} = \varnothing$; $\{0\} \vee_t \varnothing = \varnothing \vee_t \{0\} = \varnothing$
- $\{1\} \vee_t \varnothing = \varnothing \vee_t \{1\} = \{1\}$; $\{0\} \wedge_t \varnothing = \varnothing \wedge_t \{0\} = \{0\}$
- $[a,1] \wedge_t \varnothing = \varnothing \wedge_t [a,1] = \{0\}$ $(a \neq 1)$; $[a,1] \vee_t \varnothing = \varnothing \vee_t [a,1] = \{1\}$ $(a \neq 0)$
- $[a,1] \wedge_t \{0\} = \{0\}$; $[a,1] \vee_t \{0\} = [a,1]$.

Proposition 6.2. Let *B* be the lower bound interval bilattice $\boldsymbol{I^+(V)}$, then we obtain a *bt*-system by setting

$\lambda \models^* x \Leftrightarrow \lambda \in x$.

In such a system,

$Sat = I^+(V) - \{\varnothing\}$; $i(\lambda) = [\lambda, 1]$ in the case $\lambda \neq 0$; $i(0) = \{0\}$;

$Maxsat = \{\{0\}, \{1\}\}$; $Compl = \{\{0\}, \{1\}, \varnothing\}$.

Proposition 6.3. Call *normal* a valuation v assuming only values different from $\emptyset$ and $\{0\}$. Then the *bt-system* associated with $\boldsymbol{I^+(V)}$ gives the same formalisms of the graded approach to fuzzy logic provided we confine ourselves to the normal valuations.

Proof. It is possible to identify every normal valuation v with the function $v' : \mathrm{F} \rightarrow V$ such that $v(\alpha) = [v'(\alpha), 1]$ for every $\alpha \in \mathrm{F}$. Moreover, given a semantics M and $m \in \mathrm{M}$, we have that $m \models v$ provided that $m \supseteq v'$. Finally, since for every formula α,

$L_c(v)(\alpha) = [Inf\{m(\alpha) : m \supseteq v'\}, 1]$,

it is possible to identify $L_c(v)(\alpha)$ with $Inf\{m(\alpha)$: $m \supseteq v'\}$. In a similar way we can relate the deduction apparatus of the graded approach to fuzzy logic with the deduction apparatus proposed in this paper.

It is interesting to observe that, in particular,

$$Tau(\alpha) = [Inf\{m(\alpha) : m \in \mathrm{M}\}, 1].$$

Therefore, by referring to the early considered valuation structure ([0,1], $\wedge$, $\vee$, 1-x, 0, 1),

$Tau(\alpha \vee \neg\alpha) = [0.5,1]$ and $Tau(\alpha \wedge \neg\alpha) = [0,1]$.

This means that *Tau* gives no information on a contradiction.

We conclude this section with the following proposition emphasizing that $\boldsymbol{I^+(V)}$ is obtained by extending the domain V of $\boldsymbol{V}$ by two elements and the order in V into two different orders.

Proposition 6.4. Extend the domain V of $\boldsymbol{V}$ by two symbolic elements f and i (corresponding to $\{0\}$ and $\emptyset$, respectively) and set $B = V \cup \{f, i\}$. Extend the order $\leq$ in V into two orders $\leq_t$ and $\leq_k$ in such a way that

- i is a maximum with respect to $\leq_k$ and f is a minimum with respect to $\leq_t$
- $f \geq_k 0$, $i \leq_t 1$,
- f is not k-comparable with the elements in V-$\{0\}$,
- i is not t-comparable with the elements in V-$\{1\}$.

Then $(B, \leq_t, \leq_k, f, 1, 0, i)$ is a bilattice isomorphic with $\boldsymbol{I^+(V)} = (I^+(V), \leq_t, \leq_k, \{0\}, \{1\}, [0,1], \emptyset)$.

Proof. It is sufficient to consider the map $h : I^+(V) \rightarrow B$ defined by setting $h(\emptyset) = i$, $h(\{0\}) = f$ and $h([\lambda,1]) = \lambda$.

Observe that in $(B, \leq_t, \leq_k, f, 1, 0, i)$ False and True are represented by f and 1, while no information and inconsistency are represented by 0, i, respectively. Then the *bt*-system given in Proposition 6.2 gives a proper extension of Pavelka's formalisms in which we admit valuations v able to express the fact that a formula α is false (in the case $v(\alpha)$=0) and the fact that the information on α is inconsistent (in the case $v(\alpha) = i$).

7. Completeness theorem and fixed points

Let M be fuzzy semantics and (*IR*,*la*) a deduction apparatus. Then to prove a completeness theorem it is useful to examine the fixed points of the operators *D* and L_c. Indeed, (*IR*, *la*) is correct if and only if all the fixed points of L_c are fixed points of *D* and is complete if all the fixed point of *D* are fixed points of L_c. Regarding the fixed points of L_c, the general theory of the closure operators gives the following proposition.

Proposition 7.1. A valuation *v* is a fixed point of L_c if and only if *v* is a *k*-intersection of elements in $\underline{M}$. Equivalently, the set of fixed points of L_c is the closure system generated by $\underline{M}$.

Instead, we can characterize the fixed points of *D*, i.e. the theories, as the *B*-subsets of formulas closed with respect to the deduction apparatus.

Definition 7.2. Let *v* be a *B*-set of formulas, then *v* is called *closed* with respect to the *n*-ary inferential rule *r* if, for every $\alpha_1,\ldots,\alpha_n$

$$v(r_{syn}(\alpha_1,\ldots,\alpha_n)) \geq_k r_{sem}(v(\alpha_1),\ldots,v(\alpha_n)).$$

We say that *v* is *closed with respect to a fuzzy deduction apparatus* (*IR*, *la*) if *v* is closed with respect to all the inferential rules in *IR* and *v* *k*-contains the *B*-subset of logical axioms.

Observe that the closure with respect to the fusion rule is expressed by the inequality $v(\alpha) \geq_k v(\alpha) \vee_k v(\alpha)$ and therefore that all the *B*-sets of formulas are closed with respect to this rule.

Theorem 7.3. Let *v* be a valuation, then *v* is a theory (i.e. a fixed point of *D*) if and only if *v* is closed with respect to (*IR*, *la*).

Proof. Assume that *v* is closed with respect to (*IR*,*la*). To prove that *v* is a fixed point for *D*, we prove, by induction on the length of the formulas, that for every formula α and for every proof π of α

$$I(\pi,v) \leq_k v(\alpha) \qquad (7.1)$$

In the case $n = 1$, the proof consists in assuming either that α is a logical axiom or a hypothesis. In both the cases (7.1) is satisfied in a trivial way. Consider the case $n \neq 1$ and, by induction hypothesis, that (7.1) is satisfied by all the proofs whose length is less than *n*. Then again in the case α is assumed as a logical axiom or a hypothesis (7.1) holds true. Otherwise, there is an inference rule $r = (r_{syn}, r_{sem})$ such that

$$\alpha = r_{sin}(\alpha_{i(1)},\ldots, \alpha_{i(m)}) \text{ with } 1\leq i(1)<n,\ldots,1\leq i(m)<n \text{ and}$$

$$I(\pi,v) = r_{sem}(I(\pi(i(1)),v),\ldots,I(\pi(i(m)),v)).$$

Then by the closure of *v*, by induction hypothesis and the monotony of r_{sem}, we have that

$$v(\alpha)=v(r_{sin}(\pi_{i(1)},\ldots,\ \pi_{i(m)}))\geq_k r_{sem}(v(\alpha_1),\ldots,v(\alpha_n))\geq_k$$
$$\geq_k r_{sem}\,(I(\pi(i(1)),v),\ldots,I(\pi(i(m)),v))= I(\pi,v).$$

Conversely, assume that v is a fixed point of D. Then $v(\alpha) = Sup_k\{I(\pi,v) : \pi$ is a proof of $\alpha\}$ and therefore $v(\alpha) \geq_k I(\pi v)$ for every proof π of α. By assuming that π is the proof of length 1 consisting in assuming α as a logical axiom, then we obtain $v(\alpha) \geq_k I(\pi,v) = la(\alpha)$. Then v k-contains la. Let r be an n-ary inference rule, then to prove that v is closed with respect to r, given $\alpha_1,\ldots,\alpha_n$ we consider the proof π obtained by assuming $\alpha_1,\ldots,\alpha_n$ as hypotheses and by applying the rule r. Such a proof proves the formula $\alpha = r_{sin}(\alpha_1,\ldots,\alpha_n)$ and therefore

$$v(r_{sin}(\alpha_1,\ldots,\alpha_n)) = v(\alpha) \geq_k I(\pi,v) = r_{sem}(v(\alpha_1),\ldots,v(\alpha_n)).$$

Thus v is closed with respect to $(IR,\ la)$.

Definition 7.4. We say that an inference rule is *correct* with respect to a semantics M provided that, every $\underline{m}\in \underline{\mathrm{M}}$ is closed with respect to the rule.

The following simple proposition it is useful to prove the correctness of a deduction apparatus.

Proposition 7.5. A deduction apparatus (IR, la) is correct with respect to a semantics M if and only if $la \leq_k Tau$ and all the inference rule in IR are correct with respect to M.

The following simple proposition gives an useful tool to prove the completeness.

Proposition 7.6. Le ***B*** be a bilattice satisfying the decomposition property. Then to prove the completeness of the deduction apparatus it is sufficient to prove that given a theory v different from v_{T}

i) for every formula α, there is a model m_{α} of v such that

$$\underline{m}_{\alpha}(\alpha)\wedge_k True = v(\alpha)\wedge_k True$$

ii) for every formula α, there is a model m^{α} of v such that

$$\underline{m}^{\alpha}(\alpha)\wedge_k False = v(\alpha)\wedge_k False.$$

If ***B*** is with negation and both the elements in $\underline{\mathrm{M}}$ and the theories are balanced, then it is sufficient to prove *i*).

Proof. To prove that every fixed point v of D is a fixed point of L_c, observe that in the case $v = v_{\mathrm{T}}$ this is trivial. In the case $v \neq v_{\mathrm{T}}$, by the assumed hypotheses, for every formula α,

$$L_c(v)(\alpha)\wedge_k True = (Inf_k\{\underline{m}(\alpha) :\ m \models v\})\wedge_k True \leq_k \underline{m}_{\alpha}(\alpha)\wedge_k True = v(\alpha)\wedge_k True$$

and

$$L_c(v)(\alpha)\wedge_k False = (Inf_k\{\underline{m}(\alpha) :\ m \models v\})\wedge_k False \leq_k \underline{m}^{\alpha}(\alpha)\wedge_k False =$$
$$= v(\alpha)\wedge_k False.$$

Consequently,

$$L_c(v)(\alpha) = (L_c(v)(\alpha)\wedge_k True)\vee_k(L_c(v)(\alpha)\wedge_k False) \leq_k (v(\alpha)\wedge True))\vee_k(v(\alpha)\wedge_k False) = v(\alpha).$$

So, v is a fixed point of L_c and this entails the completeness.

Assume that the elements in $\underline{\mathrm{M}}$ and the fixed points of D are balanced and that *i*) holds true. Then, by *i*), given the formula $\neg\alpha$ there is a model $m_{\neg\alpha}$ such that $\underline{m}_{\neg\alpha}(\neg\alpha)\wedge_k True = v(\neg\alpha)\wedge_k True$. Consequently, if we set $m^{\alpha} = m_{\neg\alpha}$

$$\underline{m}^{\alpha}(\alpha)\wedge_k False = \sim(\sim\underline{m}^{\alpha}(\alpha)\wedge_k \sim False) = \sim(\underline{m}^{\alpha}(\neg\alpha)\wedge_k True)$$

$$= \sim(v(\neg\alpha)\wedge_k True) = (\sim v(\neg\alpha))\wedge_k(\sim True) = v(\alpha)\wedge_k False$$

and *ii*) holds true.

In the case B is a product bilattice $\boldsymbol{B(V)}$, the condition of such a proposition requires that there are two models m_{α}, m^{α} of v such that, for every formula α, $m_{\alpha}(\alpha) = v_{+}(\alpha)$ and $m^{\alpha}(\alpha) = -v_{-}(\alpha)$.

8. Boolean logic and Kripke bilattices

Now we will test our formalisms on a logic related with a Boolean truth-functional semantics. Namely, given a nonempty set W whose elements we call *worlds*, we consider the Boolean algebra $V = P(W)$ and the related product bilattice $B_W = B(P(W))$, we call *product Kripke bilattice* (see [20]). Obviously, B_W is defined by setting

$(X, Y) \leq_k (X', Y') \Leftrightarrow X \subseteq X'$ e $Y \subseteq Y'$; $(X, Y) \leq_t (X', Y') \Leftrightarrow X \subseteq X'$ e $Y \supseteq Y'$;

$\sim(X, Y) = (Y, X)$; $\bot = (\emptyset,\emptyset)$; $\mathrm{T} = (W,W)$;

$False = (\emptyset,W)$; $True = (W, \emptyset)$

The intended meaning of a valuation $v : \mathrm{F} \rightarrow B_W$ is that, for every formula α, the pair $v(\alpha) = (X,Y)$ represents:

- the set X of worlds in which the available information says that α is true
- the set Y of worlds in which the available information says that α is false.

In accordance with the formalisms proposed in Section 4, we have that a *bt*-system is defined such that, for every $X\in P(W)$ and $(A,B)\in B_W$,

$X \models^* (A,B)$ provided that $A \subseteq X$ and $B \subseteq -X$.

Moreover,

$Sat = \{(A,B) : A\cap B = \emptyset\}$; $i(X) = (X,-X)$;

$Maxsat = \{(X,-X) : X\in P(W)\}$; $Compl = \{(X,Y) : X\cup Y = W\}$.

We call *Kripke-bt-system* such a *bt*-system.

Definition 8.1. We call *Kripke truth functional semantics* the set M of mappings $m : \mathrm{F} \to P(W)$ that are truth-functional in $(P(W), \cap, \cup, -)$ i.e. such that for every $\alpha, \beta \in \mathrm{F}$,

$$m(\alpha\wedge\beta) = m(\alpha)\cap m(\beta)\ ;\ m(\alpha\vee\beta) = m(\alpha)\cup m(\beta)\ ;\ \ m(\neg\alpha) = -m(\alpha).$$

Notice that condition $m(\neg\alpha) = -m(\alpha)$ entails that the elements in $\underline{\mathrm{M}}$ are balanced. The intended meaning is that, given $m \in$ M and $\alpha\in$ F, $m(\alpha)$ is the set of worlds in which α is true.

As it is well known, we have that if α and α' are logically equivalent in classical propositional calculus then $m(\alpha) = m(\alpha')$. Moreover, $m(\alpha) = W$ for every tautology α and $m(\alpha) = \varnothing$ for every contradiction α. This entails that $L_c(v)$ is compatible with the logical equivalence and that

$Tau(\alpha) = (W, \varnothing)$ if α is a tautology

$Tau(\alpha) = (\varnothing, W)$ if α is a contradiction

$Tau(\alpha) = (\varnothing, \varnothing)$ otherwise.

To individuate a suitable inferential apparatus for the just defined semantics, at first we will give a "symmetric" version of the usual deduction apparatus in classical propositional calculus. Indeed, denote by $\alpha \to_t \beta$ the formula $\neg\alpha\vee\beta$ and by $\alpha \to_f \beta$ the formula $\beta\wedge\neg\alpha$. Then we define two rules. The *positive Modus Ponens* enables to obtain β from α and $\alpha \to_t \beta$, the *negative Modus Ponens* enables us to obtain β from α and $\alpha \to_f \beta$. We denote by MP^+ and MP^- these rules. It is evident that while MP^+ is correct in a positive sense (i.e. if α and $\alpha \to_t \beta$ are true, then β is true), MP^- is correct in a negative sense (i.e. if α and $\alpha \to_f \beta$ are false, then β is false). Also, we denote by LA one of the sets of logical axioms of classical propositional calculus and by $\neg LA$ the set $\{\neg\alpha : \alpha\in LA\}$.

Definition 8.2. We say that a set T of formulas is a *theory* or that *T is closed with respect to positive proofs* provided that T contains LA and it is closed with respect to MP^+. We say that T is an *anti-theory* or that *T is closed with respect to negative proofs* provided that T contains $\neg LA$ and it is closed with respect to MP^-.

Passing to our bilattices-based logic, we call *positive Modus Ponens* (in brief MP^+) the rule defined by setting

$$\left\langle \frac{\alpha \quad \alpha \to_t \beta}{\beta} \,\middle|\, \frac{(A_+, A_-)\ \ (I_+, I_-)}{(A_+, A_-)\lozenge^+(I_+, I_-)} \right\rangle$$

where $\lozenge^+$ is the *positive conjunction* defined by setting:

$(A_+, A_-)\lozenge^+(I_+, I_-) = (A_+ \cap I_+, \varnothing)$.

Such a rule works on the positive information since the negative components of the antecedents in this rule do not give information on the conclusion. On the dual side we can define the following rule we call *the negative Modus Ponens* (in brief *MP*$^-$),

$$\langle \frac{\alpha \quad \alpha \to_f \beta}{\beta} \mid \frac{(A_+, A_-) \quad (I_+, I_-)}{(A_+, A_-)\Diamond^-(I_+, I_-)} \rangle$$

where the *negative conjunction* $\Diamond^-$ is defined by setting

$(A_+, A_-)\Diamond^-(I_+, I_-) = (\varnothing, A_-\cap I_-)$.

In such a case the rule works only on the negative information.

Also we will consider the ¬-*elimination* and the ¬-*introduction* rules (whose meaning is obvious)

$$\langle \frac{\neg\alpha}{\alpha} \mid \frac{(X,Y)}{(Y,X)} \rangle \ ; \ \langle \frac{\alpha}{\neg\alpha} \mid \frac{(X,Y)}{(Y,X)} \rangle$$

Notice that a valuation *v* is closed with respect to these two rules if and only if *v* is balanced. Also, these rules are not independent. Finally a particular role is played by the following *inconsistency rule*

$$\langle \frac{\alpha}{\alpha} \mid \frac{(X,Y)}{k(X,Y)} \rangle$$

where the map *k* is defined by setting $k(X,Y) = (X,Y)$ if $X\cap Y = \varnothing$ and $k(X,Y) = (W,W)$ otherwise. Such a rule says that if there is a world *w* in which the information on α is inconsistent, then the information on α have to be considered inconsistent in all the worlds.

Proposition 8.3. The proposed rules satisfy the continuity condition.

Proof. To prove that $\Diamond^+$ is continuous, let $(A_i)_{i\in I}$ be a directed family of elements of B_W and $I \in B_W$. Then

$(Sup_{i\in I}(A_i^+, A_i^-))\Diamond^+(I^+, I^-) = (\cup_i A_i^+, \cup_i A_i^-)\Diamond^+(I^+, I^-) = ((\cup_i A_i^+) \cap I^+, \varnothing)$

$= (\cup_i (A_i^+\cap I^+), \varnothing) = Sup_{i\in I,}(A_i^+ \cap I^+, \varnothing)$

$= Sup_{i\in I}((A_i^+, A_i^-)\Diamond^+(I^+, I^-))$.

In a similar way one proves that $\Diamond^-$ is continuous. To prove that ~ is continuous we observe that

$\sim(Sup_{i\in I} A_i) = \sim(\cup_i A_i^+, \cup_i A_i^-) = (\cup_i A_i^-, \cup_i A_i^+) = Sup_{i\in I} \sim A_i$.

To prove that *k* is continuous, assume that $\cup_i A_{i\in I}^+$ and $\cup_{i\in I} A_i^-$ are disjoint. Then, since for every $i\in I$, A_i^+ and A_i^- are disjoint,

$k(Sup_{i\in I}A_i) = k(\cup_i A_i^+, \cup_i A_i^-) = (\cup_i A_i^+, \cup_i A_i^-) = Sup_{i\in I}(A_i^+, A_i^-) = Sup_{i\in I}k(A_i)$.

Assume that there is a word $w\in(\cup_{i\in I} A_i^+)\cap(\cup_{i\in I}A_i^-)$, then there are $\underline{i}$ and $\underline{j}$ such that $w\in A_{\underline{i}}^+\cap A_{\underline{j}}^-$. Since $(A_i)_{i\in I}$ is directed, there is A_h such that $A_{\underline{i}}^+\subseteq A_h^+$, $A_{\underline{i}}^-\subseteq A_h^-$, $A_{\underline{j}}^+\subseteq A_h^+$, $A_{\underline{j}}^-\subseteq A_h^-$, . This entails that $w\in A_h^+\cap A_h^-$ and therefore,

$k(Sup_{i\in I}A_i) = k(\cup_i A_i^+, \cup_i A_i^-) = (W, W) = k(A_{\underline{h}}) = Sup_{i\in I}k(A_i)$.

Definition 8.4. We call *Kripke deduction system*, in brief *K-system*, the deduction system in the Kripke bilattice B_W whose rules are $MP^+$and MP^- and whose B_W-set of logical axioms is defined by setting

$$la(\alpha)=\begin{cases}(\mathrm{W},\varnothing) \ \ if \ \alpha\in LA\\ (\varnothing,W) \ \ if \ \alpha\in\neg LA\\ (\varnothing,\varnothing) \ \ otherwise\end{cases}$$

It is intended that the *K*-system contains the fusion rule. The proof of the following proposition is matter of routine.

Proposition 8.5. Given a valuation *v*, the following equivalences hold true

a) $v \supseteq_k la \ \Leftrightarrow \ v_+(\alpha) = W$ and $v_-(\neg\alpha) = W$ for every $\alpha\in LA$

b) *v* is closed with respect to $MP^+ \ \Leftrightarrow \ v_+(\beta) \supseteq v_+(\alpha)\cap v_+(\alpha\rightarrow_t\beta)$ for every α and β

c) *v* is closed with respect to $MP^- \ \Leftrightarrow \ v_-(\beta) \supseteq v_-(\alpha)\cap v_-(\alpha\rightarrow_f\beta)$ for every α and β

d) *v* is closed with respect to the ¬-introduction and the ¬-elimination rules

$\Leftrightarrow v(\neg\alpha) = \sim v(\alpha) \ \Leftrightarrow \ v_+(\alpha) = v_-(\neg\alpha)$ and $v_-(\alpha) = v_+(\neg\alpha)$ for every α

e) *v* is closed with respect to the inconsistency rule ⇔ either *v* is pointwise satisfiable or $v = v_\blacklozenge$.

Notice that the sets *Sat*, *Maxsat*, *Compl*, can be also defined by the lattice operations in $\boldsymbol{B_W}$. Indeed,

$(X,Y) \in Sat \Leftrightarrow (X,Y)\wedge_t\sim(X,Y) \leq_k False$

$(X,Y) \in Maxsat \Leftrightarrow (X,Y)\vee_t\sim(X,Y) = True \Leftrightarrow (X,Y)\wedge_t\sim(X,Y) = False.$

$(X,Y) \in Compl \Leftrightarrow (X,Y)\vee_t\sim(X,Y) \geq_k True \Leftrightarrow (X,Y)\wedge_t\sim(X,Y) \geq_k False.$

As it is usual in formal logic, there is no difficulty to prove the correctness of the considered inferential apparatus.

Proposition 8.6. The *K*-system is correct with respect to the truth-functional semantics M.

Proof. It is evident that if $\underline{m}\in\underline{\mathrm{M}}$, then $\underline{m} \supseteq_k la$ and that $\underline{m}$ is closed with respect to the ¬-introduction rule, the ¬-elimination rule and the inconsistency rule. To prove that $\underline{m}$ is closed with respect to MP^+, it is sufficient to observe that

$$\underline{m}_+(\beta) = m(\beta) \supseteq m(\alpha)\cap m(\beta) = m(\alpha)\cap(m(\beta))\cup-m(\alpha)) = m(\alpha)\cap m(\beta\vee\neg\alpha) = \underline{m}_+(\alpha)\cap\underline{m}_+(\beta\vee\neg\alpha)$$

To prove that $\underline{m}$ is closed with respect to MP^-, we observe that

$$m(\beta) \subseteq m(\alpha)\cup(m(\beta)\cap-m(\alpha)) = m(\alpha)\cup m(\beta\wedge\neg\alpha)$$

and therefore that

$$\underline{m}_-(\beta) = -m(\beta) \supseteq -m(\alpha)\cap-m(\beta\wedge\neg\alpha) = \underline{m}_-(\alpha)\cap\underline{m}_-(\beta\wedge\neg\alpha).$$

9. An isomorphic bilattice

In order to find a completeness theorem relating M with the proposed *K*-system, it is useful to introduce the following bilattice.

Definition 9.1. Let P(F) be the Boolean algebra of all the subsets of F and denote by B_F the associated product bilattice $B(P(F))$. Then we call *formulas based bilattice* the bilattice B_F^W obtained as the direct power of the bilattice B_F with index set W. We call *W-valuation* the elements of such a bilattice.

Then a W-valuation $U : W \rightarrow B_F$ is defined by a pair (U_+,U_-) of functions from W into P(F) whose intended interpretation is that, for every world w,
- $U_+(w)$ is the set of formulas the available information suggests to be true in w
- $U_-(w)$ is the set of formulas the available information suggests to be false in w.
The following theorem shows that the bilattices B_W^F and B_F^W are isomorphic.

Theorem 9.2. Define the map $H : B_W^F \rightarrow B_F^W$ by setting, for every $v\in B_W^F$,

$$H(v)(w) = (T^v(w), F^v(w))$$

where,

$$T^v(w) = \{\alpha : w\in v_+(\alpha)\} \text{ and } F^v(w) = \{\alpha : w\in v_-(\alpha)\}.$$

Then H is an isomorphism between B_W^F and B_F^W whose inverse is the function $K : B_F^W \rightarrow B_W^F$ such that, for every $U\in B_F^W$ and $\alpha\in$ F,

$$K(U)(\alpha) = (\{w: \alpha\in U_+(w)\}, \{w: \alpha\in U_-(w)\}).$$

Proof. It is immediate that H and K are both one-to-one and $H^{-1} = K$. Moreover,

$u \leq_k v \Leftrightarrow$ for every $\alpha\in$ F, $u(\alpha) \leq_k v(\alpha) \Leftrightarrow$ for every $\alpha\in$ F $u_+(\alpha) \subseteq v_+(\alpha)$ and $u_-(\alpha) \subseteq v_-(\alpha) \Leftrightarrow$ for every $w\in W$,$\{\alpha : w\in u_+(\alpha)\} \subseteq \{\alpha : w\in v_+(\alpha)\}$ and $\{\alpha : w\in u_-(\alpha)\} \subseteq \{\alpha : w\in v_-(\alpha)\} \Leftrightarrow$ for every $w\in W$, $H(u)(w) \leq_k H(v)(w) \Leftrightarrow H(u) \leq_k H(v)$.

and

$u \leq_t v \Leftrightarrow$ for every $\alpha\in$ F, $u(\alpha) \leq_t v(\alpha) \Leftrightarrow$ for every $\alpha\in$ F $u_+(\alpha) \subseteq v_+(\alpha)$ and $u_-(\alpha) \supseteq v_-(\alpha) \Leftrightarrow$ for every $w\in W$, $\{\alpha : w\in u_+(\alpha)\} \subseteq \{\alpha : w\in v_+(\alpha)\}$ and $\{\alpha : w\in u_-(\alpha)\} \supseteq \{\alpha : w\in v_-(\alpha)\} \Leftrightarrow$ for every $w\in W$, $H(u)(w) \leq_t H(v)(w) \Leftrightarrow H(u) \leq_t H(v)$.

Finally,

$$H(\sim v)(w) = H((v_-, v_+))(w) = (\{\alpha : w\in v_-(\alpha)\},\{\alpha : w\in v_+(\alpha)\})$$
$$= \sim (\{\alpha : w\in v_+(\alpha)\}, \{\alpha : w\in v_-(\alpha)\}) = \sim H(v)(w).$$

Observe that, $H(v_{\perp})$ is the map constantly equal to $(\emptyset,\emptyset)$, $H(v_{T})$ the map constantly equal to (F,F) and $H(la)$ the map constantly equal to $(LA,\neg LA)$.

Definition 9.3. Let U be an element in B_{F}^{W}, then we say that

- U is *pointwise satisfiable* if, for every $w\in W$, $U_{+}(w)\cap U_{-}(w) = \emptyset$
- U is *closed with respect to MP^{+}* if, for every $w\in W$, $U_{+}(w)$ is closed with respect to MP^{+}
- U is *closed with respect to MP^{-}* if, for every $w\in W$, $U_{-}(w)$ is closed with respect to MP^{-}
- U is *balanced* if, for every $w\in W$,

$$\alpha\in U_{+}(w) \Leftrightarrow \neg\alpha\in U_{-}(w) \quad \text{and} \quad \alpha\in U_{-}(w) \Leftrightarrow \neg\alpha\in U_{+}(w).$$

- U is *complete* if, for every $w\in W$, $U_{+}(w)$ is a complete and $U_{-}(w) = -U_{+}(w)$.

Proposition 9.4. Given $v \in B_{W}^{F}$,

i) $v \supseteq_{k} la \Leftrightarrow T^{v}(w) \supseteq LA$ and $F^{v}(w) \supseteq \neg LA$

ii) v is closed with respect to $MP^{+} \Leftrightarrow T^{v}(w)$ is closed with respect to MP^{+} for every $w\in W$.

iii) v is closed with respect to $MP^{-} \Leftrightarrow F^{v}(w)$ is closed with respect to MP^{-} for every $w\in W$.

iv) v is balanced $\Leftrightarrow$ v is closed with respect to the $\neg$-elimination and $\neg$-introduction rules $\Leftrightarrow$ $H(v)$ is balanced

v) v is closed with respect to the inconsistency rule $\Leftrightarrow$ either $H(v)$ is pointwise satisfiable or $H(v)$ is constantly equal with (F,F).

Proof. Equivalences *i*), *ii*), *iii*) and *v*) are all trivial. To prove *iv*), assume that v is closed with respect to the $\neg$-introduction and $\neg$-elimination rules and therefore that, for every α, $v_{+}(\alpha) = v_{-}(\neg\alpha)$ and $v_{-}(\alpha) = v_{+}(\neg\alpha)$. Then

$$\alpha \in T^{v}(w) \Leftrightarrow w\in v_{+}(\alpha) \Leftrightarrow w\in v_{-}(\neg\alpha) \Leftrightarrow \neg\alpha\in F^{v}(w)$$

and

$$\alpha\in F^{v}(w) \Leftrightarrow w\in v_{-}(\alpha) \Leftrightarrow w\in v_{+}(\neg\alpha) \Leftrightarrow \neg\alpha\in T^{v}(w)$$

and this proves that $H(v)$ is balanced. Conversely, assume that $H(v)$ is balanced, then

$$w\in v_{+}(\alpha) \Leftrightarrow \alpha \in T^{v}(w) \Leftrightarrow \neg\alpha\in F^{v}(w) \Leftrightarrow w\in v_{-}(\neg\alpha)$$

and

$$w\in v_{-}(\alpha) \Leftrightarrow \alpha \in F^{v}(w) \Leftrightarrow \neg\alpha\in T^{v}(w) \Leftrightarrow w\in v_{+}(\neg\alpha).$$

Now, we are able to characterize the models of Kripke logic as the families of complete theories.

Proposition 9.5. Given $\underline{m}\in\underline{M}$, $H(\underline{m})$ is a complete W-valuation. Conversely, if U is a complete W-valuation, then $K(U)\in\underline{M}$. Namely, $K(U) = \underline{m}$ where m is defined by setting, for every $\alpha \in$ F, $m(\alpha) = \{w\in W : \alpha\in U_{+}(w)\}$.

Proof. Assume that $\underline{m}\in\underline{\mathrm{M}}$, then it is immediate that, for every $w\in W$, $F^{\underline{m}}(w) = -T^{\underline{m}}(w)$. Moreover, in accordance with Proposition 8.6, $\underline{m}$ is closed with respect to MP^{+} and $\underline{m}\supseteq_k la$. By Proposition 9.4 this entails that $T^{\underline{m}}(w)$ contains LA and it is closed with respect to MP^{+}. Then $T^{\underline{m}}(w)$ is a theory. To prove that $T^{\underline{m}}(w)$ is complete, observe that,

$$\alpha\in T^{\underline{m}}(w) \Leftrightarrow w\in m(\alpha) \Leftrightarrow w\notin m(\neg\alpha) \Leftrightarrow \neg\alpha\notin T^{\underline{m}}(w).$$

Conversely, assume that for every $w\in W$, $U_+(w)$ is a complete and that $U_-(w) = -U_+(w)$ and define m by setting $m(\alpha) = \{w\in W : \alpha\in U_+(w)\}$. Then m is truth-functional. Indeed, since $U_+(w)$ is closed under deductions,

$w\in m(\gamma\wedge\beta) \Leftrightarrow \gamma\wedge\beta\in U_+(w) \Leftrightarrow \gamma\in U_+(w)$ and $\beta\in U_+(w)$

$\Leftrightarrow w\in m(\gamma)$ and $w\in m(\beta) \Leftrightarrow w\in m(\gamma)\cap m(\beta)$.

Then $m(\gamma\wedge\beta) = m(\gamma)\cap m(\beta)$. Moreover, since $U_+(w)$ is complete

$w\in m(\gamma\vee\beta) \Leftrightarrow \gamma\vee\beta\in U_+(w) \Leftrightarrow \gamma\in U_+(w)$ or $\beta\in U_+(w)$

$\Leftrightarrow w\in m(\gamma)$ or $w\in m(\beta) \Leftrightarrow w\in m(\gamma)\cup m(\beta)$.

This means that $m(\gamma\vee\beta) = m(\gamma)\cup m(\beta)$. Finally,

$w\in m(\neg\gamma) \Leftrightarrow \neg\gamma\in U_+(w) \Leftrightarrow \gamma\notin U_+(w) \Leftrightarrow w\notin m(\gamma)$

and this proves that $m(\neg\gamma) = -m(\gamma)$.

On the other hand, since for every $w\in W$,

$w\in -m(\alpha) \Leftrightarrow \alpha\notin U_+(w) \Leftrightarrow \alpha\in U_-(w)$,

we have also

$\underline{m}(\alpha) = (m(\alpha), -m(\alpha)) = (\{w: \alpha\in U_+(w)\}, \{w: \alpha\in U_-(w)\}) = K(U)(\alpha)$.

Corollary 9.6. Given a valuation v, a model m of v exists if and only if there is a family $(T_w)_{w\in W}$ of complete theories such that $T^v(w)\subseteq T_w\subseteq -F^v(w)$. The model m is obtained by setting, for every formula α, $m(\alpha) = \{w\in W : \alpha\in T_w\}$.

Proof. It is evident that if a model m of v exists, then, $H(\underline{m})$ is a complete W-valuation and, since $\underline{m}\geq_k v$, $H(\underline{m})\geq_k H(v)$. This entails that $(H_+(\underline{m})(w))_{w\in W}$ is a family of complete theories such that $T^v(w)\subseteq H_+(\underline{m})(w)\subseteq -F^v(w)$.

Conversely, consider a family of complete theories $(T_w)_{w\in W}$ such that $T^v(w)\subseteq T_w\subseteq -F^v(w)$. Then we can consider the W-valuation U obtained by setting $U_+(w) = T_w$ and $U_-(w) = -T_w$. By definition U is complete and therefore by setting $m(\alpha) = \{w\in W : \alpha\in T_w\}$ we obtain an element m of M such that $\underline{m} = K(U)$. Since by hypothesis $T^v(w)\subseteq T_w$ and $-T_w\supseteq F^v(w)$, we have $\underline{m} = K(U)\geq_k v$, i.e. $\underline{m}\models v$.

Proposition 9.7. Given a valuation $v\in B_W^{\mathrm{F}}$, $v\neq v_{\mathrm{T}}$, the following are equivalents:

i) v is a theory

ii) $T^v(w)$ is a consistent theory for every $w\in W$ and $H(v)$ is balanced.

iii) $F^v(w)$ is a consistent anti-theory for every $w\in W$ and $H(v)$ is balanced.

Proof. The implications *i*) ⇒ *ii*) and *i*) ⇒ *iii*) are evident. To prove that *ii*) ⇒ *i*) we observe that $H(v)$ is pointwise satisfiable and therefore v is closed with respect to the inconsistency rule. Indeed, assume that there are $\alpha\in\mathrm{F}$ and $w\in W$ such that $\alpha\in T^v(w)\cap F^v(w)$. Then, since $H(v)$ is balanced, $\neg\alpha\in T^v(w)$ and therefore $T^v(w)$ is inconsistent. This contradict the hypothesis $T^v(w)\neq\mathrm{F}$. To prove the closure of v with respect to MP^- we prove that $F^v(w)$ is closed with respect to MP^-. Now if α and $\alpha\rightarrow_f\beta$ are in $F^v(w)$, then $\neg\alpha$ and $\neg(\beta\wedge\neg\alpha)\in T^v(w)$. Since $T^v(w)$ is a theory, this means that $\neg\alpha$ and $\neg\alpha\rightarrow_t\neg\beta\in T^v(w)$ and therefore $\neg\beta\in T^v(w)$. Thus, since $H(v)$ is balanced, we can conclude that $\beta\in F^v(w)$. In a similar way one proves that *iii*) ⇒ *i*).

10. The completeness theorem

In Section 8 we proved the correctness of the considered inferential apparatus. Taking in account the results of Section 9 we are ready to prove the completeness, too.

Proposition 10.1. The *K*-system is complete with respect to the truth-functional semantics M.

Proof. Since both the elements in $\underline{M}$ and the fixed points of D are balanced, by Proposition 7.6 it is sufficient to prove that, for every formula α, there is a model m_α of v such that $m_\alpha(\alpha) = v_+(\alpha)$. Now, since v is fixed point of D, for every $w\in W$, $T^v(w)$ is a consistent theory that $T^v(w)\supseteq\neg F^v(w)$ and $F^v(w)\supseteq\neg T^v(w)$. Define $U_\alpha = (U_+^\alpha, U_-^\alpha)\in B_{\mathrm{F}}^{\,W}$ by setting:

- $U_+^\alpha(w)$ equal to any complete theory extending $T^v(w)$ in the case $\alpha\in T^v(w)$
    - $U_+^\alpha(w)$ equal to any complete theory extending $T^v(w)\cup\{\neg\alpha\}$ in the case $\alpha\notin T^v(w)$
- $U_-^\alpha(w) = -U_+^\alpha(w)$.

Then $U_+^\alpha(w)$ is a complete extension of $T^v(w)$ such that, trivially,

$$\alpha\in U_+^\alpha(w) \Leftrightarrow \alpha\in T^v(w).$$

Now, by Corollary 9.6, U_α is associated with an element $\underline{m}_\alpha$ in $\underline{M}$. Namely, $\underline{m}_\alpha$ is defined by the model $m_\alpha\in\mathrm{M}$ such that $m_\alpha(\beta) = \{w : \beta\in U_+^\alpha(w)\}$ for every formula β. To prove that m_α is a model of v, we observe that, by definition, $U_+^\alpha(w)\supseteq T^v(w)$. To prove that $U_+^\alpha(w)\subseteq -F^v(w)$ we observe that, for every $\beta\in U_+^\alpha(w)$, $\neg\beta\notin U_+^\alpha(w)$ and therefore $\neg\beta\notin T^v(w)$. So $\beta\notin F^v(w)$. This proves that m_α is a model of v. To prove that $m_\alpha(\alpha) = v_+(\alpha)$, we observe that

$$w\in m_\alpha(\alpha) \Leftrightarrow \alpha\in U_+^\alpha(w) \Leftrightarrow \alpha\in T^v(w) \Leftrightarrow w\in v_+(\alpha).$$

We can summarize the results in this section as follows.

Theorem 10.2. The Kripke truth-functional semantics and the Kripke deduction system define a bilattice-based fuzzy logic.

We conclude such a section by emphasizing the algebraic features of the fixed points of *D*. Indeed, denote by F/≡ the Lindenbaum algebra of the propositional calculus and for every valuation *v* set

$$[T^v(w)] = \{[\alpha] \in \mathrm{F}/\!\equiv\, : \alpha \in T^v(w)\} \text{ and } [F^v(w)] = \{[\alpha] \in \mathrm{F}/\!\equiv\, : \alpha \in F^v(w)\}.$$

Then if v is a fixed point of *D*, $[T^v{}_w]$ is a proper filter and $[F^v{}_w]$ the corresponding dual ideal in F/≡. If $v = \underline{m}$ with $\underline{m} \in \underline{\mathrm{M}}$, then $[T^v{}_w]$ is maximal and $[F^v{}_w]$ is its complement.

11. Inconsistency-tolerant Kripke logic

In the just considered fuzzy bilattice logic there is no tolerance with respect to the inconsistency of the information. Indeed, assume that in a given valuation v there is a formula α such that $v_+(\alpha) \cap v_-(\alpha) \neq 0$. Then in such a logic no model exists for v and therefore $L_c(v) = v_\blacklozenge$. So, the whole information content of v is useless. This is disturbing since the utility of bilattice theory is also to manage inconsistency. In alternative, we can attempt to consider the following inconsistency-tolerant logic. At first we observe that given a bilattice *B*, we can obtain a *bt*-system by assuming that the valuation structure *V* coincides with *B*, in a sense.

Proposition 11.1. Given a bilattice $\boldsymbol{B}$, we obtain a *bt*-system $(\boldsymbol{V}, \boldsymbol{B}, \models^*)$ by setting $\boldsymbol{V} = (B, \leq_t)$ and $\lambda \models^* x \Leftrightarrow \lambda \geq_k x$.

In such a system

$Sat = B$; $Maxsat = Compl = \{\mathrm{T}\}$; $i(\lambda) = \lambda$.

Moreover, if $\mathrm{M} \subseteq V^{\mathrm{F}}$ is a semantics, for every $m \in \mathrm{M}$

$m \models v$ provided $\Leftrightarrow m \sqsupseteq_k v$,

and therefore,

$L_c(v)(\alpha) = Inf_k\{m(\alpha) : m \sqsupseteq_k v\}$.

It is evident that in such a case the set *Sat, Maxsat* and *Compl* are meaningless. Also, we are interested in the following semantics in B_W.

Definition 11.2. Let M be the Kripke truth-functional semantics given in Definition 8.1 and, for every $m\in \mathrm{M}$ and $X\subset W$, define $\underline{m}_X$ by setting $\underline{m}_X = \underline{m}\vee_k(X,X)$. Then the *inconsistency-tolerant Kripke semantics* is defined by setting $\underline{\mathrm{M}}_{inc} = \{\underline{m}_X : X\subset W, m\in \mathrm{M}\}$.

Observe that if α is a tautology, then $\underline{m}_X(\alpha) = (W, X)$, if α is a contradiction, then $\underline{m}_X(\alpha) = (X, W)$. Moreover, $\underline{m}_X$ is compatible with the logical equivalence.

Proposition 11.3. Let $v : \mathrm{F} \to B_W$ be an initial valuation and set $I = \cup_{\alpha\in F} v_+(\alpha)\cap v_-(\alpha)$. Then $\underline{m}_X \models v$ only if $X\supseteq I$. Moreover, for every $X\subset W$, denote by v^X the valuation defined by setting $v^X(\alpha) = (v_+(\alpha)-X, v_-(\alpha)-X)$ for every $\alpha\in \mathrm{F}$. Then the following are equivalent:

i) $\underline{m}_X \models v$

ii) $\underline{m}_X \models v^I$, $X \supseteq I$.

iii) $\underline{m} \supseteq_k v^X$

Consequently,

$$L_c(v) = \cap_k\{\underline{m}_X : m\in \mathrm{M}, I \subseteq X\subset W, \underline{m} \supseteq_k v^X\} \tag{11.1}$$

and therefore the B_W-set of tautologies coincides with the one defined by M.

Proof. If $\underline{m}_X \models v$ then for every $\alpha\in \mathrm{F}$, $m(\alpha)\cup X \supseteq v_+(\alpha)$ and $-m(\alpha)\cup X \supseteq v_-(\alpha)$ and therefore

$X = (m(\alpha)\cup X)\cap(-m(\alpha)\cup X) \supseteq v_+(\alpha)\cap v_-(\alpha)$.

This entails that $X \supseteq I$.

i) $\Rightarrow$ *ii)*. Assume that $\underline{m}_X \models v$, then it is evident that $\underline{m}_X \models v^I$ and $X\supseteq I$.

ii) $\Rightarrow$ *iii)*. Assume that $\underline{m}_X \models v^I$ with $X\supseteq I$, then, $\underline{m}_X \supseteq_k v^I$ and therefore for every $\alpha\in \mathrm{F}$, $m(\alpha)\cup X \supseteq v_+(\alpha)-I$ and $(-m(\alpha))\cup X \supseteq v_-(\alpha)$-$I$. In turn this entails $m(\alpha) \supseteq (v_+(\alpha)$-$I)-X$ and $-m(\alpha) \supseteq (v_-(\alpha)$-$I)-X$ and therefore $m(\alpha) \supseteq v_+(\alpha)-X$ and $-m(\alpha) \supseteq v_-(\alpha)-X$. So, $\underline{m} \supseteq_k v^X$.

iii) $\Rightarrow$ *i)* Observe that

$\underline{m} \supseteq_k v^X \Rightarrow \underline{m}_X \models v \Rightarrow$ for every $\alpha\in \mathrm{F}$, $m(\alpha) \supseteq v_+(\alpha)-X$ and $-m(\alpha) \supseteq v_-(\alpha)-X \Rightarrow$ for every $\alpha\in \mathrm{F}$, $m(\alpha)\cup X \supseteq v_+(\alpha)$ and $-m(\alpha)\cup X \supseteq v_-(\alpha) \Rightarrow \underline{m}_X \models v$.

Then, even if there is α such that $v_+(\alpha)\cap v_-(\alpha) \neq \emptyset$, it is again possible that v admits a model. This since we can search for models $\underline{m}_X$ of the pointwise consistent valuation v^I such that $X \supseteq I$.

Definition 11.4. We call *inconsistency-tolerant Kripke deduction system*, in brief *t-K-system*, the deduction system obtained from the *K-system* by deleting the inconsistency rule. We denote by D_t the related deduction operator.

Observe that the inconsistency rule is deleted since is useless. Indeed the function k is the identity.

The proof of the following proposition is similar with the one of Proposition 9.7.

Proposition 11.5. Given a valuation $v \in B_W^{\mathrm{F}}$, the following are equivalent:

i) v is a fixed point of D_t

ii) $T^v(w)$ is a theory for every $w \in W$ and $H(v)$ is balanced.

iii) $F^v(w)$ is an anti-theory for every $w \in W$ and $H(v)$ is balanced.

Proposition 11.6. The *t-K*-system is correct with respect to the semantics $\underline{\mathrm{M}}_{inc}$.

Proof. Let $\underline{m}_X = \underline{m} \vee_k (X,X) \in \underline{\mathrm{M}}_{inc}$ then $\underline{m}_X \supseteq \underline{m} \supseteq_k la$. Moreover, since

$\underline{m}_X(\neg\alpha) = (m(\neg\alpha)\cup X, -m(\neg\alpha)\cup X) = ((-m(\alpha))\cup X, m(\alpha)\cup X) = \sim\underline{m}_X(\alpha)$.

$\underline{m}_X$ is closed with respect to the $\neg$-introduction and the $\neg$-elimination rules. To prove that $\underline{m}_X$ is closed with respect to MP^+, it is sufficient to observe that

$$X\cup m(\beta) \supseteq X\cup(m(\alpha)\cap m(\beta)) = X\cup(m(\alpha)\cap(m(\beta)\cup -m(\alpha)))$$

$$= X\cup(m(\alpha)\cap m(\beta\vee\neg\alpha)) = (m(\alpha)\cup X)\cap(m(\beta\vee\neg\alpha)\cup X).$$

To prove that $\underline{m}_X$ is closed with respect to MP^-, we observe that

$$X\cup -m(\beta) \supseteq X\cup(-m(\alpha)\cap -m(\beta)) = X\cup(-m(\alpha)\cap(-m(\beta)\cup m(\alpha)))$$

$$= X\cup(-m(\alpha)\cap -(m(\beta)\cap -m(\alpha))) = X\cup(-m(\alpha)\cap -m(\beta\wedge\neg\alpha))$$

$$= (X\cup -m(\alpha))\cap(X\cup -m(\beta\wedge\neg\alpha)).$$

Proposition 11.7. The *t-K-system* is complete with respect to the semantics $\underline{\mathrm{M}}_{inc}$.

Proof. Since both the elements in $\underline{\mathrm{M}}_{inc}$ and the fixed points of D_t are balanced, by Proposition 7.6 it is sufficient to prove that, if v is a fixed point of D_t different from v_{T} then for every formula α, there is a model $\underline{n}_\alpha \in \underline{\mathrm{M}}_{inc}$ of v such that the first component of $\underline{n}_\alpha(\alpha)$ is $v_+(\alpha)$. Now, since v is a fixed point of D_t, by Proposition 11.5, given $w \in W$ either $T^v(w) = \mathrm{F}$ or $T^v(w)$ is a consistent theory closed under deductions. We set

$X = \{w \in W : T^v(w) = \mathrm{F}\} = \cap_{\gamma\in F} v_+(\gamma)$.

It is immediate that since $v \neq v_{\mathrm{T}}$, $X \neq W$. Also, we define $U_\alpha = (U_+^{\alpha}, U_-^{\alpha}) \in B_{\mathrm{F}}^{W}$ by setting:

- $U_+^{\alpha}(w)$ equal to any complete theory if $w \in X$
- $U_+^{\alpha}(w)$ equal to any complete theory extending $T^v(w)$ if $w \notin X$ and $\alpha \in T^v(w)$
- $U_+^{\alpha}(w)$ equal to any complete theory extending $T^v(w)\cup\{\neg\alpha\}$ if $w \notin X$ and $\alpha \notin T^v(w)$
- $U_-^{\alpha}(w) = -U_+^{\alpha}(w)$ for every $w \in W$.

It is evident that, for every $w \notin X$, $U_+^{\alpha}(w)$ is a complete extension of $T^v(w)$ such that,

$\alpha \in U_+^{\alpha}(w) \Leftrightarrow \alpha \in T^v(w)$.

Now, by Proposition 9.5, if we define m_α by setting $m_\alpha(\beta) = \{w : \beta \in U_+^\alpha(w)\}$, then m_α is an element of M such that $H(\underline{m}_\alpha) = U_\alpha$. We claim that $\underline{m}_\alpha \vee_k (X,X)$ is a model of v, i.e., by Proposition 11.3, that, for every $\beta \in$ F,

$$m_\alpha(\beta) \supseteq v_+(\beta)\text{-}X \text{ and } \text{-}m_\alpha(\beta) \supseteq v_-(\beta)\text{-}X. \qquad (11.2)$$

To this aim, at first we observe that, for every $w \notin X$

$$(U_+^\alpha(w), U_-^\alpha(w)) \geq_k (T^v(w), F^v(w)). \qquad (11.3)$$

Indeed, by definition, $U_+^\alpha(w) \supseteq T^v(w)$. To prove that $U_-^\alpha(w) \supseteq F^v(w)$ observe that, since v is closed with respect to the $\neg$-introduction and the $\neg$-elimination rule, $T^v(w) \supseteq \neg F^v(w)$ and $F^v(w) \supseteq \neg T^v(w)$. Then, for every $\beta \in F^v(w)$, since $\neg\beta \in T^v(w)$, it is also $\neg\beta \in U_+^\alpha(w)$. In turn, since $U_+^\alpha(w)$ is consistent, this entails that $\beta \notin U_+^\alpha(w)$ and therefore $\beta \in U_-^\alpha(w)$. Coming back to (11.2), assume that $w \in v_+(\beta)\text{-}X$, then $\beta \in T^v(w)$ and therefore, since $w \notin X$, $\beta \in U_+^\alpha(w)$. Then $w \in m_\alpha(\beta)$. Assume that
$w \in v_-(\beta)\text{-}X$, then $\beta \in F^v(w)$ and therefore $\beta \in U_-^\alpha(w)$. Then $\beta \notin U_+^\alpha(w)$ and this entails that $w \in \text{-}m_\alpha(\beta)$.

To prove that the first component of $(\underline{m}_\alpha \vee_k (X,X))(\alpha)$ is $v_+(\alpha)$, i.e. that

$$w \in m_\alpha(\alpha) \cup X \Leftrightarrow w \in v_+(\alpha),$$

we observe that such an equivalence is evident in the case $w \in X$. Otherwise

$$w \in m_\alpha(\alpha) \Leftrightarrow \alpha \in U_+^\alpha(w) \Leftrightarrow \alpha \in T^v(w) \Leftrightarrow w \in v_+(\alpha).$$

We can summarize the results in this section as follows.

Theorem 11.8. The semantics $\underline{\mathrm{M}}_{inc}$ and the *t*-*K*-deduction system define a bilattice-based fuzzy logic.

12. Extending the Kripke bilattice logic

Perhaps it is possible to extend the just considered logic related to *Kripke*-*bt*-systems to obtain similar logics in any *bt*-system. Even we will consider such a question in a future work, in this section we sketch some ideas and results.

Definition 12.1. Given any *bt*-system $(\boldsymbol{V}, \boldsymbol{B}, \models^*)$ in a bilattice $\boldsymbol{B}$ with a negation, the *canonical semantics associated with* $(\boldsymbol{V}, \boldsymbol{B}, \models^*)$ is the semantics defined by the class of maps $\underline{n} : F \to B$ which are *B-truth-functional* in $(B, \wedge_t, \vee_t, \sim)$, *i. e.* such that

$$\underline{n}(\alpha\wedge\beta) = \underline{n}(\alpha)\wedge_t \underline{n}(\beta) \;;\; \underline{n}(\alpha\vee\beta) = \underline{n}(\alpha)\vee_t \underline{n}(\beta) \;;\; \underline{n}(\neg\alpha) = \sim\underline{n}(\alpha)$$

and such that, for every formula α, $\underline{n}(\alpha)$ is complete and satisfiable.

The following proposition shows that such a definition extends the one of Kripke semantics.

Proposition 12.2. Let M be the Kripke semantics. Then, given $\underline{n} : \mathrm{F} \to B_W$, the following are equivalent:

i) $\underline{n}\in\underline{\mathrm{M}}$

ii) $\underline{n}$ is B_W-truth-functional and its values are satisfiable and complete

iii) $\underline{n}$ is B_W-truth-functional and its values in the propositional variables are satisfiable and complete.

Therefore, M is the canonical semantics associated with the Kripke-*bt*-system.

Proof. i) ⇒ *ii*) Assume that $\underline{n} \in \underline{\mathrm{M}}$, i.e. that there is $m\in\mathrm{M}$ such that $\underline{n} = i\circ m$. It is immediate that the values assumed by $\underline{n}$ are complete and satisfiable. Moreover

$$\underline{n}(\alpha\wedge\beta) = (m(\alpha\wedge\beta),-m(\alpha\wedge\beta)) = (m(\alpha)\cap m(\beta),-(m(\alpha)\cap m(\beta))) =$$

$$=(m(\alpha)\cap m(\beta),-m(\alpha)\cup -m(\beta)) = \underline{n}(\alpha)\wedge_t\underline{n}(\beta).$$

$$\underline{n}(\alpha\vee\beta) = (m(\alpha\vee\beta),-m(\alpha\vee\beta)) = (m(\alpha)\cup m(\beta),-(m(\alpha)\cup m(\beta)))$$

$$= (m(\alpha)\cup m(\beta),-m(\alpha)\cap -m(\beta)) = \underline{n}(\alpha)\vee_t\underline{n}(\beta).$$

$$\underline{n}(\neg\alpha) = (m(\neg\alpha), -m(\neg\alpha)) = (-m(\alpha), m(\alpha)) = {\sim}\underline{n}(\alpha).$$

ii) ⇒ *iii*) Evident.

iii) ⇒*i*) Let $\underline{n} : \mathrm{F} \to B_W$ a B_W-truth functional valuation whose values in the propositional variables are complete and satisfiable. Let m the element of M defined in a truth-functional way by assigning to every propositional variable p_i the value $m(p_i) = \underline{n}_+(p_i)$. We claim that $\underline{n}$ coincides with $\underline{m}$. Indeed, since $\underline{n}$ is truth functional by hypothesis and $\underline{m}$ is truth functional by implication *i*)⇒*ii*), it is sufficient to prove that $\underline{n}(p_i) = \underline{m}(p_i)$ for every propositional variable p_i. On the other hand, since $\underline{n}(p_i)$ is satisfiable and complete,

$$\underline{n}(p_i) = (n_+(p_i), n_-(p_i)) = (n_+(p_i) , -n_+(p_i)) = \underline{m}(p_i).$$

Definition 12.3. Given any *bt*-system (***V***, ***B***, ⊨*) in a bilattice ***B*** with a negation, we call *canonical deduction apparatus associated with* (***V***, ***B***, ⊨*) the deduction apparatus (*IR*, *la*) such that *la* is defined by setting, for every $\alpha\in\mathrm{F}$,

$$la(\alpha) = \begin{cases} True \;\; if \;\; \alpha\in LA \\ False \;\; if \;\; \alpha\in\neg LA \\ \bot \;\; otherwise \end{cases}$$

and *IR* is the set of the following inference rules

$$\langle\frac{\alpha \quad \alpha\to_t\beta}{\beta} \mid \frac{\lambda \qquad \mu}{\lambda\wedge_k\mu\wedge_k True}\rangle \quad \text{(Positive Modus Ponens)}$$

$$\langle\frac{\alpha \quad \alpha\to_f\beta}{\beta} \mid \frac{\lambda \qquad \mu}{\lambda\wedge_k\mu\wedge_k False}\rangle \quad \text{(Negative Modus Ponens)}$$

$$\langle\frac{\neg\alpha}{\alpha} \mid \frac{\lambda}{\sim\lambda}\rangle \qquad \text{(}\neg\text{-elimination)}$$

$\langle\frac{\alpha}{\neg\alpha}\mid\frac{\lambda}{\sim\lambda}\rangle$ (¬-introduction)

$\langle\frac{\alpha}{\alpha}\mid\frac{\lambda}{k(\lambda)}\rangle$ (consistency)

where $k(\lambda) = \lambda$ if $\lambda \in Sat$ and $k(\lambda) = \mathrm{T}$, otherwise.

Obviously, some trivial hypotheses on ***B*** are necessary to obtain the continuity condition for the inference rules. The following proposition shows that such a definition extends the one of *K-system*.

Proposition 12.4. The *K-system* is the canonical deduction apparatus associated with the Kripke-*bt*-system.

Proof. We observe only that

$(A_+, A_-)\Diamond^-(I_+, I_-) = (A_+, A_-)\wedge_k(I_+, I_-)\wedge_k True$ and

$(A_+, A_-)\Diamond^-(I_+, I_-) = (A_+ , A_-)\wedge_k(I_+, I_-)\wedge_k False$.

We can give similar definitions by referring to the inconsistency-tolerant Kripke logic. Indeed, it is possible to consider the same deduction apparatus apart the inconsistency rule and the semantics suggested by *iii*) of the following proposition.

Definition 12.5. Given any *bt*-system (***V***, ***B***, $\models$*) in a bilattice ***B*** with a negation, the *canonical inconsistency-tolerant semantics associated with* (***V***, ***B***, $\models$*) is the semantics defined by the class of maps $\underline{n} : F \to B$ which are *B-truth-functional* in $(B, \wedge_t, \vee_t, \sim)$ and such that there is $c \neq \mathrm{T}$ such that, for every formula α, $\underline{n}(\alpha)$ is complete and $\underline{n}(\alpha)\wedge_k{\sim}\underline{n}(\alpha) = c$

Proposition 12.6. Let $\underline{\mathrm{M}}_{inc}$ be the inconsistency-tolerant Kripke semantics and $\underline{n} : \mathrm{F} \to B_W$ be a map, then the following are equivalent:

i) $\underline{n} \in \underline{\mathrm{M}}_{inc}$

ii) $\underline{n}$ is B_W-truth-functional and there is $c \neq \mathrm{T}$ such that, for every formula α, $\underline{n}(\alpha)$ is complete and $\underline{n}(\alpha)\wedge_k{\sim}\underline{n}(\alpha) = c$

iii) $\underline{n}$ is B_W-truth-functional and there is $c \neq \mathrm{T}$ such that, for every propositional variable p_i the value of $\underline{n}(p_i)$ is complete and $\underline{n}(p_i)\wedge_k{\sim}\underline{n}(p_i) = c$.

Consequently, Definition 12.5 extends the notion of inconsistency-tolerant Kripke semantics.

Proof. i) $\Rightarrow$ *ii*) Assume that $\underline{n} \in \underline{\mathrm{M}}_{inc}$ and therefore that there is $m \in \mathrm{M}$ and $X \subset W$ such that $\underline{n} = \underline{m}_X$, then, for every formula α,

$$\underline{n}(\alpha)\wedge_k\sim\underline{n}(\alpha) = (m(\alpha)\cup X,-m(\alpha)\cup X)\wedge_k(-m(\alpha)\cup X, m(\alpha)\cup X) =$$

$$= ((m(\alpha)\cup X)\cap(-m(\alpha)\cup X), (m(\alpha)\cup X)\cap(-m(\alpha)\cup X) = (X,X)$$

where $(X,X) \neq (W,W)$. It is evident that $\underline{n}$ is pointwise complete. To prove that $\underline{n}$ is B_W-truth-functional, we observe that, since B_W is distributive,

$$\underline{n}(\alpha\wedge\beta) = \underline{m}(\alpha\wedge\beta)\vee_k(X,X) = (\underline{m}(\alpha)\wedge_t\underline{m}(\beta))\vee_k(X,X) =$$

$$= (\underline{m}(\alpha)\vee_k(X,X))\wedge_t(\underline{m}(\beta)\vee_k(X,X)) = \underline{n}(\alpha)\wedge_t\underline{n}(\beta).$$

Likewise,

$$\underline{n}(\alpha\vee\beta) = \underline{m}(\alpha\vee\beta)\vee_k(X,X) = (\underline{m}(\alpha)\vee_t\underline{m}(\beta))\vee_k(X,X) =$$

$$= (\underline{m}(\alpha)\vee_k(X,X))\vee_t(\underline{m}(\beta)\vee_k(X,X)) = \underline{n}(\alpha)\vee_t\underline{n}(\beta).$$

Finally,

$$\underline{n}(\neg\alpha) = \underline{m}(\neg\alpha)\vee_k(X,X) = \sim\underline{m}(\alpha)\vee_k(X,X) = ((-m(\alpha))\cup X, m(\alpha)\cup X) =$$

$$= \sim(\underline{m}(\alpha)\vee_k(X,X)) = \sim\underline{n}(\alpha).$$

ii) $\Rightarrow$ *iii*) Evident.

iii) $\Rightarrow$ *i*) Let $\underline{n} : \mathrm{F} \rightarrow B_W$ a B_W-truth functional valuation such that, for every propositional variable p_i, $\underline{n}(p_i)$ is complete and $\underline{n}(p_i)\wedge\sim\underline{n}(p_i) = (X,X)$, $X\neq W$. Let m the element of M defined in a truth-functional way by assigning to every propositional variable p_i the value $m(p_i) = n_+(p_i)-X$. We claim that $\underline{n}$ coincides with $\underline{m}_X$. Indeed, given a propositional variable p_i, by hypothesis $n_+(p_i)\cup n_-(p_i) = W$ and $n_+(p_i)\cap n_-(p_i) = X$ and therefore, since $\{n_+(p_i)-X, n_-(p_i)\}$ is a partition of W, $n_-(p_i) = -(n_+(p_i)-X)$. Then

$$\underline{n}(p_i) = (n_+(p_i), n_-(p_i)) = ((n_+(p_i)-X)\cup X, n_-(p_i)\cup X) =$$

$$= ((n_+(p_i)-X)\cup X ,\ -(n_+(p_i)-X)\cup X) = \underline{m}(p_i)\vee_k(X,X) = \underline{m}_X(p_i).$$

Now, $\underline{n}$ is truth functional by hypothesis and $\underline{m}_X$ is truth-functional by implication *i*) $\Rightarrow$ *ii*). Then the fact that $\underline{n}$ coincides with $\underline{m}_X$ in the propositional variables entails that $\underline{n} = \underline{m}_X$.

13. About the meaning of the canonical deduction system

To give an idea of the meaning of a canonical deduction apparatus, assume that ***B*** is the product *bt*-system ***B***(***V***) defined in Proposition 5.2 and G is a set of *generators* of V, i.e. a set of truth values such that for every $\lambda\in V$, $\lambda = sup\{g\in G \mid g\leq\lambda\}$. For example, if V is the real numbers interval [0,1] we can put G equal to the set of rational numbers in [0,1]. If V is the Boolean algebra B_W, we can assume that G is the set of singletons. Under these conditions, we can consider a bilattice similar to the one considered in Section 9, namely the bilattice $B_{\mathrm{F}}{}^G$ obtained as the direct power of B_{F} with index set G. Also, we can associate every valuation v with the family $H(v) = (T^v(\lambda), F^v(\lambda))_{\lambda\in G}$ where $T^v(\lambda)$ and $F^v(\lambda)$ are the *positive* λ*-cut and negative* λ*-cut of* v defined by

$$T^v(\lambda) = \{\alpha \mid v_+(\alpha) \geq \lambda\} \text{ and } F^v(\lambda) = \{\alpha \mid v_-(\alpha) \geq \lambda\}.$$

Then if all the elements in G are prime, H is an algebraic homomorphism $H : B^{\mathrm{F}} \rightarrow B_{\mathrm{F}}{}^{G}$ from the bilattice B^{F} into $B_{\mathrm{F}}{}^{G}$. Since

$v(\alpha) = (sup\{g\in G \mid \alpha\in\ T^{v}(\lambda)\}, sup\{g\in G \mid \alpha\in\ F^{v}(\lambda)\})$,

such a homomorphism is an embedding.

If, for example we consider the the canonical inconsistency-tolerant system. Then the fixed points of the related deduction operator are the valuations v such that, for every $\alpha, \beta \in \mathrm{F}$,

i) $v(\alpha) \geq_k True$ for every $\alpha\in LA$

ii) $v(\alpha) \geq_k False$ for every $\alpha\in \neg LA$

iii) $v(\beta) \geq_k v(\alpha)\wedge_k v(\alpha \rightarrow_t \beta)\wedge_k True$

iv) $v(\beta) \geq_k v(\alpha)\wedge_k v(\alpha \rightarrow_f \beta)\wedge_k False$

v) $v(\neg\alpha) = \sim v(\alpha)$,

i.e. such that

i) $v_+(\alpha) = 1$ for every $\alpha\in LA$

ii) $v_-(\alpha) = 1$ for every $\alpha\in \neg LA$

iii) $v_+(\beta) \geq v_+(\alpha)\wedge v_+(\alpha \rightarrow_t \beta)$

iv) $v_-(\beta) \geq v_-(\alpha)\wedge v_-(\alpha \rightarrow_f \beta)$

v) $v_+(\neg\alpha) = v_-(\alpha)$; $v_-(\neg\alpha) = v_+(\alpha)$

In turn, it is evident that v satisfies these conditions if and only if, for every $\lambda\in G$,

- $T^{v}(\lambda)$ is a theory
- $F^{v}(\lambda)$ is an anti-theory
- $\alpha\in F^{v}(\lambda) \Leftrightarrow \neg\alpha \in T^{v}(\lambda)$; $\alpha\in T^{v}(\lambda) \Leftrightarrow \neg\alpha \in F^{v}(\lambda)$.

Thus, a canonical inconsistency-tolerant system is able to generate a family $(T^{v}(\lambda), F^{v}(\lambda))_{\lambda\in G}$ where $T^{v}(\lambda)$ is the set of formulas we can prove at degree λ and $F^{v}(\lambda)$ is the set of formulas we can disprove at degree λ.

14. Remarks

The just exposed logics are related with the Kripke bilattice logics proposed by Ginsberg. Indeed, in [20] a valuation satisfying *ii*) of Proposition 11.5 is called *W-closed*. Moreover, the *W-closure* of a valuation v is defined as the k-intersection of all the W-closed valuations k-containing v. Consequently, in accordance with such a proposition, the W-closed valuations coincide with the fixed points of the deduction operator D_t and $D_t(v)$ coincides with the W-closure of v. Finally, Ginsberg characterizes the W-closed valuations in theoretic bilattice terms by showing that a valuation v is W-closed if and only if

1. if β is a consequence of α, then $v(\alpha) \leq_t v(\beta)$
2. $v(\alpha\wedge\beta) \geq_k v(\alpha)\wedge_t v(\beta)$

3. $v(\neg\alpha) = \sim v(\beta)$.

In accordance with the fact that the *W*-closed valuations coincide with the fixed points of D_t, this paper gives a further characterization of the *W*-closed valuations by the conditions

i) $v(\alpha) \geq_k True$ for every $\alpha \in LA$;

ii) $v(\alpha) \geq_k False$ for every $\alpha \in \neg LA$;

iii) $v(\beta) \geq_k v(\alpha) \wedge_k v(\alpha \rightarrow_t \beta) \wedge_k True$;

iv) $v(\beta) \geq_k v(\alpha) \wedge_k v(\alpha \rightarrow_f \beta) \wedge_k False$ *;*

v) $v(\neg\alpha) = \sim v(\alpha)$.

This means that our formalisms give a semantics and a deduction system in Pavelka's style (as in the tradition of logic) for Ginsberg's notion of *W*-closure.

It is evident that several open questions exist. As an example, an open question is to find suitable conditions on a *bt*-system to obtain that the associated canonical semantics and canonic deduction apparatus are related by a completeness theorem. Also, perhaps it is interesting to investigate about the connections of these logics with the notions of necessity and possibility in fuzzy set theory (see [17]). Indeed, by referring to the product *bt*-system considered in Section 13, we have that the positive part of a theory *v* is a theory of the generalized necessity logic proposed in [17].

However, the main open question is that, in spite of the possible interest of the logics proposed in Sections 8-13, every serious investigation about the connection between fuzzy logic and bilattice theory leads to face up with the valuation structures usually considered in many-valued logic (see for example [6]). While suggestions to connect these structures with bilattice theory are in [8], as far as we know these connections are not investigate in the framework of formal logic.

CHAPTER 5

PRESERVATION THEOREMS

1. The cuts of a fuzzy structure

A natural modification of a fuzzy structure is to transform it into a crisp structure by a cutting operation. This is done everytime one decides that truth values beyond a given level are sufficient to claim that a vague property is satisfied. A precise definition is the following.

Definition 1.1. Let 0 and 1 be the minimum and the maximum in a given valuation structure $\boldsymbol{V}$ and $\lambda\in V$. Then the function $c_\lambda : V \rightarrow\{0,1\}$ is defined by setting

$c_\lambda(x) = 1$ if $x \geq \lambda$

$c_\lambda(x) = 0$ otherwise.

Given a fuzzy subset $s : S \rightarrow V$, the *λ-cut* of s is the crisp fuzzy subset $s_\lambda = c_\lambda \circ s$.

Equivalently, the λ-cut s_λ is the characteristic function of the subset

$$C(s,\lambda) = \{x\in S : s(x) \geq\lambda\}.$$

As usual, we identify s_λ with $C(s,\lambda)$. A fuzzy subset s is completely determined by the associate family $(s_\lambda)_{\lambda\in V}$ of its cuts. Indeed, we have that, for every $x\in S$,

$$s(x) = sup_{\lambda\in V}\ \lambda\wedge s_\lambda(x) = sup\{\lambda \in V : x\in C(s,\lambda)\}.$$

Observe that the family $(C(s,\lambda))_{\lambda\in V}$ is *continuous*, i.e.

$$C(s,\mu) = \cup_{\lambda<\mu}C(s,\lambda).$$

Definition 1.2. Given a fuzzy structure M and $\lambda\in V$, the *λ-cut of* M is the interpretation $M_\lambda = (D, \{0,1\}, I_\lambda)$ in which the constants and the operation symbols are interpreted as in M and such that, for every relation symbol $\underline{r}$, $I_\lambda(\underline{r})$ is the λ-cut of $I(\underline{r})$, i.e. , for every $d_1,...,d_n$ in D

$$I_\lambda(\underline{r})(d_1,...,d_n) = c_\lambda(I(\underline{r})(d_1,...,d_n)).$$

Equivalently we can define M_λ as the classical structure with the same algebraic structure as M and in which

$$I_\lambda(\underline{r}) = \{(d_1,...,d_n) : I(\underline{r})(d_1,...,d_n) \geq \lambda\}.$$

It is evident that a fuzzy structure is completely determined by the family of its cuts. Also, we can characterize the homomorphism from M to M' by referencing to the cuts of these structures.

Proposition 1.3. Let $M = (D, V, I)$ and $M' = (D',V',I')$ be two fuzzy structures, h a homomorphism from ***Al***(M) into ***Al***(M') and k a homomorphism from ***VAL***(***M***) into ***VAL***(***M'***). Then

(h, k) is a weak homomorphism

$\Leftrightarrow$ h is a weak homomorphism from M_λ to $M'_{k(\lambda)}$ for every $\lambda \in V$.

Assume that k is an isomorphism, then

(h, k) is a homomorphism

$\Leftrightarrow$ h is a homomorphism from M_λ to $M'_{k(\lambda)}$ for every $\lambda \in V$.

Proof. To prove the first part of the proposition, assume that (h,k) is a weak homomorphism, let $\underline{r}$ be a relation symbol and $(d_1,\ldots,d_n) \in D^n$. Then,

$$(d_1,\ldots,d_n) \in C(I(\underline{r}),\lambda) \Leftrightarrow I(\underline{r})(d_1,\ldots,d_n) \geq \lambda \Rightarrow k(I(\underline{r})(d_1,\ldots,d_n)) \geq k(\lambda)$$

$$\Rightarrow I'(\underline{r})(h((d_1,\ldots,d_n)) \geq k(\lambda)$$

$$\Leftrightarrow h(d_1,\ldots,d_n) \in C(I'(\underline{r}), k(\lambda)).$$

This proves that h is a weak homomorphism from M_λ to $M'_{k(\lambda)}$. Vice versa assume that, for every $\lambda \in V$, h is a weak homomorphism from M_λ to $M'_{k(\lambda)}$. Then, by setting $\lambda = I(\underline{r})(d_1,\ldots,d_n)$, since $(d_1,\ldots,d_n) \in C(I(\underline{r}),\lambda)$, it is $h(d_1,\ldots,d_n) \in C(I'(\underline{r}), k(\lambda))$ and therefore

$$I'(\underline{r})(h((d_1,\ldots,d_n)) \geq k(\lambda) = k(I(\underline{r})(d_1,\ldots,d_n)).$$

This proves that (h,k) is a weak homomorphism from M to M'.

To prove the second part of the proposition, assume that k is an isomorphism. Then, in the case (h,k) is a homomorphism,

$$(d_1,\ldots,d_n) \in C(I(\underline{r}),\lambda) \Leftrightarrow I(\underline{r})(d_1,\ldots,d_n) \geq \lambda \Leftrightarrow k(I(\underline{r})(d_1,\ldots,d_n)) \geq k(\lambda)$$

$$\Leftrightarrow I'(\underline{r})(h(d_1,\ldots,d_n)) \geq k(\lambda)$$

$$\Leftrightarrow h(d_1,\ldots,d_n) \in C(I'(\underline{r}), k(\lambda)).$$

This proves that h is a homomorphism from M_λ to $M'_{k(\lambda)}$.

Conversely, assume that h is a homomorphism from M_λ to $M'_{k(\lambda)}$ for every $\lambda \in V$. Then, since we have just proved that (h,k) is a weak homomorphism, we have only to prove that $I'(\underline{r})(h(d_1,\ldots,d_n)) \leq k(I(\underline{r})(d_1,\ldots,d_n))$. Let λ be an element in V such that $k(\lambda) = I'(\underline{r})(h(d_1,\ldots,d_n))$, then $h(d_1,\ldots,d_n) \in C(I'(\underline{r}), k(\lambda))$ and therefore $(d_1,\ldots,d_n) \in C(I(\underline{r}),\lambda)$, i.e. $I(\underline{r})(d_1,\ldots,d_n) \geq \lambda$. Then,

$$k(I(\underline{r})(d_1,\ldots,d_n)) \geq k(\lambda) = I'(\underline{r})(h(d_1,\ldots,d_n)).$$

We can describe the connection among a fuzzy model and the associated family of cuts in terms of category theory. The objects of such a category are defined as follows.

Definition 1.4. A *continuous chain of first order structures* is a family $(M_\lambda)_{\lambda\in V}$ of crisp first order interpretations of a given language with the same domain D and such that, $(I_\lambda(\underline{r}))_{\lambda\in V}$ is a continuous chain in D^n for every n-ary relation $\underline{r}$.

The morphisms of our category are defined as follows.

Definition 1.5. A *weak morphism* from a continuous chain $(M_\lambda)_{\lambda\in V}$ into a continuous chain $(M'_\lambda)_{\lambda\in V'}$ is a pair (h,k) such that k is a homomorphism from ***VAL*(**M) into ***VAL*(**M'), and h is a weak homomorphism from M_λ to $M'_{k(\lambda)}$ for every $\lambda\in V$.

Definition 1.6. The *category of continuous chains of first order structures* is the category whose objects are the continuous chains of first order structures and whose morphisms are the weak morphisms given in Definition 1.5.

To proof of the following proposition is immediate.

Proposition 1.7. Let H be the map associating every fuzzy structure M with the related family $(M_\lambda)_{\lambda\in V}$ of λ-cuts and every morphism (h,k) with (h,k). Then H is a functor from the category of fuzzy structure into the category of continuous chains of first order structures.

Proposition 1.8. Let K be the map associating every continuous chain of first order structures $(M_\lambda)_{\lambda\in V}$ with the fuzzy structure (D,I) whose algebraic part coincides with the common algebraic part of $(M_\lambda)_{\lambda\in V}$ and such that, for every n-ary relation symbol

$$I(\underline{r})(d_1,\ldots,d_n) = Sup_{\lambda\in V} I_\lambda(\underline{r})\ (d_1,\ldots,d_n).$$

Also, assume that K associate every morphism (h, k) with (h, k). Then K is a functor from the category of the continuous chains of first order structures into the category of the fuzzy structures. Moreover, K is the inverse of H.

2. Properties preserved by a cut

Usually a fuzzy structure does not satisfy the same first order properties of its cuts. Nevertheless there is an important class of fuzzy formulas for which this holds true. To show this, at first we emphasize some properties of the map c_λ.

Proposition 2.1. The function c_λ is monotone with respect to λ and therefore

$$c_\lambda(x\otimes y) \leq c_\lambda(x)\otimes c_\lambda(y). \qquad (2.1)$$

Assume that λ is idempotent, then

$$c_\lambda(x\otimes y) = c_\lambda(x)\otimes c_\lambda(y). \qquad (2.2)$$

Proof. It is immediate that c_λ is monotone and therefore that, since $x\otimes y \leq x$ and $x\otimes y \leq y$, $c_\lambda(x\otimes y) \leq c_\lambda(x)$ and $c_\lambda(x\otimes y) \leq c_\lambda(y)$. So, $c_\lambda(x\otimes y)\leq c_\lambda(x)\otimes c_\lambda(y)$. Let λ be idempotent, then to prove the inequality $c_\lambda(x\otimes y) \geq c_\lambda(x)\otimes c_\lambda(y)$ we observe that

$$c_\lambda(x)\otimes c_\lambda(y) = 1 \;\Rightarrow\; x \geq \lambda \text{ and } y \geq \lambda \;\Rightarrow\; x\otimes y \geq \lambda\otimes\lambda = \lambda \Rightarrow\; c_\lambda(x\otimes y) = 1.$$

we will consider the atomic formulas.

Proposition 2.2. Let M be a fuzzy structure and $\lambda\in V$. Then, for every atomic formula α,

$$val(M_\lambda,\alpha, d_1,...,d_n) = c_\lambda(val(M,\alpha, d_1,...,d_n)) \qquad (2.3)$$

Consequently, given $\mu\in V$,

$$M \vDash <\alpha,\mu> \;\Leftrightarrow\; M_\lambda \vDash \alpha \text{ for every } \lambda \leq \mu \qquad (2.4)$$

Proof. To prove (2.2) observe that if $\alpha = \underline{r}(t_1,\ldots,t_m)$, then

$$val(M_\lambda, \alpha, d_1,...,d_n) = I_\lambda(\underline{r})(I_\lambda(t_1)(d_1,...,d_n),..., I_\lambda(t_m)(d_1,...,d_n))$$
$$= c_\lambda(I(\underline{r})(I(t_1)(d_1,...,d_n),..., I(t_m)(d_1,...,d_n)))$$
$$= c_\lambda(val(M,\alpha,d_1,...,d_n)).$$

To prove (2.4) assume that M satisfies $<\alpha, \mu>$ and therefore that $val(M,\alpha, d_1,...,d_n) \geq \mu$ for every $d_1,\ldots,d_n$ in D. Then

$$val(M_\lambda,\alpha,d_1,...,d_n) = c_\lambda(val(M,\alpha,d_1,...,d_n)) \geq c_\lambda(\mu) = 1$$

and this proves that M_λ satisfies α. Assume that $M_\lambda \vDash \alpha$ and therefore that $c_\lambda(val(M,\alpha, d_1,...,d_n)) = 1$ for every $\lambda\leq\mu$. Then $val(M,\alpha, d_1,...,d_n) \geq \lambda$ for every $\lambda\leq\mu$ and this proves that $val(M,\alpha, d_1,...,d_n) \geq \mu$.

Theorem 2.3. Let M be a fuzzy structure and let $<\alpha, \mu>$ be a positive fuzzy clause. Then

$$M \models <\alpha,\mu> \Rightarrow M_\lambda \models \alpha. \text{ for every idempotent } \lambda \text{ such that } \lambda \leq \mu. \qquad (2.5)$$

In the case all the elements λ in V are idempotent, then

$$M \models <\alpha,\mu> \ \Leftrightarrow \ M_\lambda \models \alpha \text{ for every } \lambda \leq \mu \qquad (2.6)$$

Proof. In the case α is an atomic formula both (2.5) and (2.6) are immediate consequences of Proposition 2.2. Assume that α is the positive clause $\forall x_1...\forall x_n(\beta_1\underline{\triangle}^*...\ \underline{\triangle}^*\beta_t \underline{\rightarrow} \beta)$ and that $M \models <\alpha,\mu>$. Then,

$$val(M, \beta_1, d_1,...,d_n)\otimes...\otimes val(M, \beta_t, d_1,...,d_n)\otimes\mu \leq val(M, \beta, d_1,...,d_n)$$

and therefore, since $c_\lambda(\mu) = 1$ and λ is idempotent, by (2.2)

$$val(M_\lambda, \beta_1, d_1,...,d_n))\otimes...\otimes val(M_\lambda, \beta_t, d_1,...,d_n) \leq val(M_\lambda,\beta,d_1,...,d_n)$$

and this proves that $M_\lambda \models \alpha$.

To prove (2.6), assume that all the elements in V are idempotent and that $M_\lambda \models \alpha$ for every $\lambda \leq \mu$. Then, for every $d_1,...,d_n \in D$,

$$val(M_\lambda, \beta_1, d_1,...,d_n)\otimes...\otimes val(M_\lambda, \beta_t, d_1,...,d_n) \leq val(M_\lambda, \beta, d_1,...,d_n)$$

and therefore, by (2.2),

$$c_\lambda(val(M, \beta_1, d_1,...,d_n))\otimes...\otimes c_\lambda(val(M, \beta_t, d_1,...,d_n)) \leq c_\lambda(val(M, \beta, d_1,...,d_n)).$$

Set $\lambda = val(M, \beta_1, d_1,...,d_n)\otimes...\otimes val(M, \beta_t, d_1,...,d_n)\otimes\mu$, then $\lambda\leq\mu$ and

$$c_\lambda(val(M, \beta_1, d_1,...,d_n)) = \ ... = c_\lambda(val(M, \beta_t, d_1,...,d_n)) = 1.$$

Consequently $c_\lambda(val(M, \beta, d_1,...,d_n)) = 1$ and therefore

$$val(M, \beta, d_1,...,d_n) \geq \lambda = val(M, \beta_1, d_1,...,d_n)\otimes...\otimes val(M, \beta_t, d_1,...,d_n)\otimes\mu.$$

Thus, $M \models <\alpha,\mu>$.

Corollary 2.4. Assume that all the elements λ in V are idempotent, let P be a positive program and M be a fuzzy structure. Then

$$M \models P \ \Leftrightarrow \ M_\lambda \models P \text{ for every } \lambda \in V. \qquad (2.7)$$

Consequently, the maps H and K defined at end of Section 1 defines two functors from the category of fuzzy models of P into the category of continuous chain of models of P.

In particular, in the case all the elements λ in V are idempotent, given an algebraic structure $\boldsymbol{A}$ and a fuzzy subset s of $\boldsymbol{A}$,

$(\boldsymbol{A},s)$ is a fuzzy subalgebra $\Leftrightarrow$ all the cuts of s are subalgebras of $\boldsymbol{A}$

Moreover, the functors H and K defined in Section 1 enable us to identify a fuzzy subsgroup with a continuous chain of fuzzy subgroups. Likewise, if S is a set and e a binary fuzzy relation in S, then

(S, e) is a similarity $\Leftrightarrow$ all the cuts of e are equivalence relations.

Moreover we can identify a similarity with a continuous chain of of equivalence relations.

3. Connecting valuations and homomorphisms

It is possible to extend to fuzzy logic some preserving theorems of classical first order logic. To do this, we consider particular classes of formulas. We say that a logical connective is *positive in the valuation structure* $\boldsymbol{V}$, if its interpretation in $\boldsymbol{V}$ is an order preserving function. A matrix is *positive in* $\boldsymbol{V}$**,** if is defined only by positive connectives in $\boldsymbol{V}$. An universal formula is *positive in* $\boldsymbol{V}$ provided that its matrix is positive in $\boldsymbol{V}$. We say that a formula α is *identity-free* in the case there is no occurrence in α of the identity symbol =.

Proposition 3.1. Let (h,k) be a weak homomorphism from M to M'. Then the following claims hold true.

i) For every matrix α which is positive with respect to $\boldsymbol{V}'$ and $d_1,\dots,d_n \in D$

$$k(val(M, \alpha, d_1,\dots,d_m)) \leq val(M', \alpha, h(d_1),\dots,h(d_m)) \qquad (3.1)$$

ii) Let h be surjective and let α be an universal formula which is positive with respect to $\boldsymbol{V}'$ and such that both $val(M, \alpha)$ and $val(M', \alpha)$ are defined. Then

$$k(val(M, \alpha)) \leq val(M', \alpha) \qquad (3.2)$$

Proof. We prove *i*) by induction on the complexity of α. Indeed, if α is the atomic formula $\underline{r}(t_1,\dots,t_n)$, then

$$\begin{aligned} &k(val(M, \underline{r}(t_1,\dots,t_n), d_1,\dots,d_m)) \\ &\quad = k(I(\underline{r})(I(t_1)(d_1,\dots,d_m),...,I(t_n)(d_1,...,d_m)) \\ &\quad \leq I'(\underline{r})(h(I(t_1)(d_1,\dots,d_m)),...,h(I(t_n)(d_1,...,d_m))) \\ &\quad = I'(\underline{r})(I'(t_1)(h(d_1),\dots,h(d_m)),...,I'(t_n)(h(d_1),...,h(d_m))) \\ &\quad = val(M', \underline{r}(t_1,\dots,t_n), h(d_1),\dots,h(d_m)) \end{aligned}$$

and this proves that α satisfies (3.1). Assume that (3.1) is satisfied by $\alpha_1,\dots,\alpha_n$ and let $\underline{c}$ be an n-ary logical connective which is positive. Then

$$\begin{aligned}
&k(val(M\,,\underline{c}(\alpha_1,\dots,\alpha_n), d_1,\dots,d_m)\\
&\qquad = k(I(\underline{c})(val(M, \alpha_1, d_1,\dots,d_m),\dots,val(M, \alpha_n,d_1,\dots,d_m)))\\
&\qquad = I'(\underline{c})(k(val(M\,, \alpha_1, d_1,\dots,d_m)),\dots, k(val(M, \alpha_n, d_1,\dots,d_m)))\\
&\qquad \leq I'(\underline{c})(val(M', \alpha_1, h(d_1),\dots, h(d_m)),\dots,val(M', \alpha_n, h(d_1),\dots,h(d_m)))\\
&\qquad = val(M', \underline{c}(\alpha_1,\dots,\alpha_n), h(d_1),\dots,h(d_m))
\end{aligned}$$

and this proves that (3.1) is satisfied by $\underline{c}(\alpha_1,\dots,\alpha_n)$.

To prove *ii*), let $\alpha = \forall x_1...,\forall x_m(\beta)$ where β is a positive matrix, then

$$\begin{aligned}
k(val(M, \alpha)) &= k(Inf\{val(M, \beta, d_1,\dots,d_m) : d_1 \in D,\dots,d_m \in D\})\\
&\leq Inf\{k(val(M, \beta, d_1,\dots,d_m)) : d_1 \in D,\dots,d_m \in D\}\\
&\leq Inf\{val(M', \beta, h(d_1),\dots,h(d_m)) : d_1 \in D,\dots,d_m \in D\}\\
&= Inf\{val(M', \beta, d'_1,\dots,d'_m) : d'_1 \in D',\dots,d'_m \in D'\} = val(M', \alpha).
\end{aligned}$$

Definition 3.2. A homomorphism (h,k) from M to M' is called *inf-preserving in weak sense* provided that, for every formula α such that both $Inf(V(\alpha)))$ and $Inf\, k(V(\alpha))$ exist,

$$k(Inf(V(\alpha))) = Inf\, k(V(\alpha)).$$

We say that (h,k) is *inf-preserving* provided that k is *inf-preserving*, i.e.

$$k(Inf(X)) = Inf\, k(X)$$

for every subset X of V such that both $Inf(X)$ and $Inf\, k(X)$ exist.

Proposition 3.3. Let (h,k) be a homomorphism from M to M' and let α be a formula. Then the following claims hold true.

i) If α is an identity-free matrix, then for every $d_1,\dots,d_n \in D$

$$val(M', \alpha, h(d_1),\dots,h(d_m)) = k(val(M\,, \alpha, d_1,\dots,d_m)) \qquad (3.3)$$

ii) If k is inf-preserving in a weak sense and α is an identity-free universal formula, then if both $val(M\,,\alpha)$ and $k(val(M, \alpha))$ exist,

$$val(M', \alpha) \leq k(val(M, \alpha)) \qquad (3.4)$$

iii) Assume that k be inf-preserving in a weak sense and that h is surjective, then if α is any identity-free formula such that both $val(M', \alpha, h(d_1),\dots,h(d_m))$ and $val(M, \alpha, d_1,\dots,d_m)$ exist, then

$$val(M', \alpha, h(d_1),\dots,h(d_m)) = k(val(M, \alpha, d_1,\dots,d_m)). \qquad (3.5)$$

Proof. The proof of *i*) is an obvious modification of the proof of *i*) in Proposition 3.3. To prove *ii*), assume that $\alpha = \forall x_1...,\forall x_m(\beta)$ where β is an identity-free matrix . Then

$$\begin{aligned} k(val(M, \alpha)) &= k(Inf\{val(M, \beta, d_1,\ldots,d_m) : d_1 \in D,\ldots,d_m \in D\}) \\ &= Inf\{k(val(M, \beta, d_1,\ldots,d_m)) : d_1 \in D,\ldots,d_m \in D\} \\ &= Inf\{val(M', \beta, h(d_1),\ldots,h(d_m)) : d_1 \in D,\ldots,d_m \in D\} \\ &\geq Inf\{val(M', \beta, d'_1,\ldots,d'_m) : d'_1 \in D',\ldots,d'_m \in D'\} = val(M', \alpha). \end{aligned}$$

To prove *iii*) we observe that, by (3.3), equation (3.5) is satisfied by all the atomic formulas. Assume that (3.5) is satisfied by $\alpha_1,\ldots,\alpha_n$ and let $\underline{c}$ be an *n*-ary logical connective. Then

$$\begin{aligned} & k(val(M , \underline{c}(\alpha_1,\ldots,\alpha_n),d_1,\ldots,d_m) \\ &\quad = k(I(\underline{c})(val(M ,\alpha_1, d_1,\ldots,d_m),\ldots,val(M ,\alpha_n, d_1,\ldots,d_m))) \\ &\quad = I'(\underline{c})(k(val(M , \alpha_1, d_1,\ldots,d_m)),\ldots, k(val(M , \alpha_n, d_1,\ldots,d_m))) \\ &\quad = I'(\underline{c})(val(M', \alpha_1, h(d_1),\ldots, h(d_m)),\ldots,val(M', \alpha_n, h(d_1),\ldots,h(d_m))) \\ &\quad = val(M', \underline{c}(\alpha_1,\ldots,\alpha_n), h(d_1),\ldots,h(d_m)). \end{aligned}$$

This proves that (3.5) is satisfied by $\underline{c}(\alpha_1,\ldots,\alpha_n)$. Assume that (3.5) is satisfied by β. Then

$$\begin{aligned} k(val(M, \forall x_i(\beta), d_1,\ldots,d_m)) &= k(Inf\{val(M, \beta, d_1,\ldots,d_{i-1},d,d_{i+1},...,d_m) : d \in D\}) \\ &= Inf\{k(val(M, \beta, d_1,\ldots,d_{i-1},d,d_{i+1},...,d_m)) : d \in D\} \\ &= Inf\{val(M', \beta, h(d_1),\ldots, h(d_{i-1}), h(d), h(d_{i+1}),..., h(d_m)) : d \in D)\} \\ &= Inf\{val(M', \beta, h(d_1),\ldots, h(d_{i-1}), d', h(d_{i+1}),..., h(d_m)), d' \in D'\} \\ &= val(M', \alpha, h(d_1),..., h(d_m)). \end{aligned}$$

This proves that (3.5) it is satisfied by $\forall x_i(\beta)$.

An obvious extension of the proof of Proposition 3.3 enables us to prove the following theorem emphasizing that every pair of isomorphic fuzzy structures are "*elementary equivalent*".

Theorem 3.4. Let (*h*, *k*) be an isomorphism between two save fuzzy structures *M* and *M'*. Then, for every formula α and $d_1,\ldots,d_n \in D$,

$$k(val(M , \alpha, d_1,\ldots,d_m)) = val(M', \alpha, h(d_1),\ldots,h(d_m)) \qquad (3.9)$$

In particular, for every closed formula α,

$$k(val(M, \alpha)) = val(M', \alpha) \qquad (3.10)$$

As an example the map *k* defined in Example 1 is a valuation isomorphism between the fuzzy subgroups M_{16} and M'_{16} and this entails that these fuzzy structures are logically equivalent. Notice that if in the valuation structure we consider also the operation ~ defined by setting $\sim x = 1-x$, then k is not an isomorphism at all. Indeed $k(\sim 0.5) = k(0.5) = 0.9$ while $\sim k(0.5) = 0.1$. This is in accordance with the fact that a formula as $\exists x(\underline{s}(x) \leftrightarrow \underline{\neg}\ \underline{s}(x))$ is satisfied in M_{16} and it is not satisfied in M'_{16}.

Proposition 3.5. Let (h, k) be a homomorphism from M to M' such that h is surjective and let α be an universal formula which is either identity-free or positive, then

$$val(M, \alpha) = 1 \Rightarrow val(M', \alpha) = 1 \qquad (3.11)$$

Proof. Assume that $\alpha = \forall x_1 ...,\forall x_m(\beta)$ and that $val(M, \alpha) = 1$, i.e. that $val(M, \beta, d_1,\dots,d_m)) = 1$ for every $d_1,\dots,d_m \in D$. Then in the case α is positive by (3.2) in Proposition 3.1

$$val(M', \alpha) \geq k(val(M, \alpha)) \geq k(1) = 1.$$

In the case α is identity-free, since by (3.3)

$$val(M', \beta, h(d_1),\dots,h(d_m)) = k(val(M, \beta, d_1,\dots,d_m)),$$

we have

$$val(M', \alpha,) = Inf\{val(M', \beta, h(d_1),\dots,h(d_m)), d_1,\dots,d_m \in D\}$$

$$= Inf\{k(val(M, \beta, d_1,\dots,d_m)) : d_1,\dots,d_m \in D \} = 1.$$

4. Preservation theorems for fuzzy formulas

We can reformulate the results in Section 2 in terms of fuzzy properties preserved by homomorphisms.

Proposition 4.1. Let (h, k) be a weak homomorphism from M to M' with h surjective. Then, for every universal fuzzy formula $<\alpha,\lambda>$ which is positive with respect to $\boldsymbol{V'}$ and such that both $val(M', \alpha)$ and $val(M, \alpha)$ are defined,

$$M \vDash <\alpha,\lambda> \Rightarrow M' \vDash <\alpha, k(\lambda)> \qquad (4.1)$$

Proof. Assume that $M \vDash <\alpha,\lambda>$ and therefore that $val(M, \alpha) \geq \lambda$. Then, by *ii*) of Proposition 3.1, $val(M', \alpha) \geq k(val(M, \alpha)) \geq k(\lambda)$ and this proves that $M' \vDash <\alpha, k(\lambda)>$.

Proposition 4.2. Let (h, k) be a homomorphism from M to M'. Then

i) if k is injective and inf-preserving in a weak sense, then, for every identity-free universal fuzzy formula $<\alpha,\lambda>$ such that both $val(M', \alpha)$ and $val(M, \alpha)$ are defined,

$$M' \vDash <\alpha, k(\lambda)> \Rightarrow M \vDash <\alpha,\lambda> \qquad (4.2)$$

ii) if k is injective and inf-preserving in a weak sense and h is surjective, then, for every identity-free fuzzy formula $<\alpha,\lambda>$ such that both $val(M', \alpha)$ and $val(M, \alpha)$ are defined,

$$M \vDash <\alpha,\lambda> \Leftrightarrow M' \vDash <\alpha, k(\lambda)> \qquad (4.3)$$

Proof. To prove *i*), assume that $M' \vDash <\alpha, k(\lambda)>$. Then, by (3.4), $k(val(M, \alpha)) \geq val(M', \alpha) \geq k(\lambda)$. Since k is injective, this entails that $val(M, \alpha) \geq \lambda$ and therefore that $M \vDash <\alpha,\lambda>$.

To prove *ii*), observe that, by *iii*) in Proposition 3.2,

$$M \vDash <\alpha,\lambda> \Leftrightarrow val(M, \alpha) \geq \lambda \Leftrightarrow k(val(M, \alpha)) \geq k(\lambda) \Leftrightarrow val(M', \alpha) \geq k(\lambda)$$

$$\Leftrightarrow M' \vDash <\alpha, k(\lambda)>.$$

Theorem 4.3. Let (h, k) be an isomorphism from the safe structure M into the safe structure to M'. Then, for every fuzzy formula $<\alpha,\lambda>$,

$$M \vDash <\alpha,\lambda> \Leftrightarrow M' \vDash <\alpha, k(\lambda)>. \qquad (4.4)$$

As an immediate consequence of Proposition 3.6 we obtain:

Proposition 4.4. Let (h, k) be a homomorphism from M to M' with h surjective and let α be an universal formula which is either identity-free or positive in V' and such that both $val(M', \alpha)$ and $val(M, \alpha)$ are defined. Then

$$M \vDash \alpha \Rightarrow M' \vDash \alpha \qquad (4.5)$$

5. Quotients and preservation theorems

Notice that if $\equiv$ is a congruence in a valuation structure $\boldsymbol{V}$, then every logical connective which is positive in $\boldsymbol{V}$ is positive in the quotient $\boldsymbol{V}/\equiv$, too. Consequently, every formula α which is positive in $\boldsymbol{V}$ is positive in $\boldsymbol{V}/\equiv$.

Proposition 5.1. Let $\equiv$ be a congruence in a fuzzy structure M. Then the following claims hold true.

i) If α is a positive matrix, then for every $d_1,\ldots,d_n \in D$

$$[val(M, \alpha, d_1,\ldots,d_m)] \leq val(M/\equiv, \alpha, [d_1],\ldots,[d_m]). \qquad (5.1)$$

ii) If α is a positive universal formula α such that both $val(M, \alpha)$ and $val(M/\equiv, \alpha)$ exist, then,

$$[val(M, \alpha)] \leq val(M/\equiv, \alpha). \qquad (5.2)$$

iii) If α is an identity-free matrix, then, for every $d_1,\ldots,d_n \in D$,

$$[val(M, \alpha, d_1,\ldots,d_m)] = val(M/\equiv, \alpha, [d_1],\ldots,[d_m]). \qquad (5.3)$$

Proof. Claims *i*), *ii*) and *iii*) are consequences of *i*) and *ii*) of Proposition 3.1 and *i*) of Proposition 3.3, respectively.

The following is an immediate consequence of iii) of Proposition 3.2.

Theorem 5.2. Let $\equiv$ be a congruence in a fuzzy structure M whose canonical homomorphism is inf-preserving in a weak sense. Then, for every identity-free formula α,

$$val(M/\equiv, \alpha, [d_1],\ldots,[d_m]) = [val(M, \alpha, d_1,\ldots,d_m)]. \qquad (5.4)$$

There is a simple characterization of the congruences whose canonical homomorphism is inf-preserving.

Proposition 5.3. Let $\equiv$ be a congruence in a valuation structure ***V***, then the associated canonical homomorphism is *inf*-preserving if and only if all the complete classes are closed with respect to the *inf* operator.

Proof. Assume that the canonical homomorphism is inf-preserving and assume that $(\lambda_i)_{i\in I}$ is a family of elements in a class $[c]$. Then since $[Inf(\lambda_i)_{i\in I}] = Inf_{i\in I}\,[\lambda_i] = [c]$, $Inf(\lambda_i)_{i\in I}$ is in the class $[c]$. Conversely, assume that all the complete classes are closed with respect to the inf operator, then since $Inf(\lambda_i)_{i\in I} \leq \lambda_i$ it is also $[Inf(\lambda_i)_{i\in I}] \leq [\lambda_i]$ and this shows that $[Inf(\lambda_i)_{i\in I}]$ is a lower bound of the family $([\lambda_i])_{i\in I}$. Let $[m]$ be a lower bound of such a family. Then, for every $i\in I$, $m\wedge\lambda_i \equiv m$. Consequently, $Inf_{i\in I}\, m\wedge\lambda_i \equiv m$ and, since $Inf_{i\in I}\, m\wedge\lambda_i = m\wedge(Inf_{i\in I}\,\lambda_i)$ it is also $m\wedge(Inf_{i\in I}\,\lambda_i) \equiv m$. In turn this implies that $[m] \leq [Inf(\lambda_i)_{i\in I}]$. Thus, $[Inf(\lambda_i)_{i\in I}] = Inf_{i\in I}\,[\lambda_i]$.

Proposition 5.4. Let α be an universal formula which is either identity-free or positive in V and $\equiv$ be a congruence in M, then

$$M \vDash \alpha \Rightarrow M/\equiv \vDash \alpha \qquad (5.5)$$

Such a theorem entails, for example, that the quotient of a fuzzy subgroup is a fuzzy subgroup. Indeed, the fuzzy subgroups are the models of a fuzzy theory whose formulas are either equations (which are positive universal formulas) or identity-free formulas. Again, it entails that the quotient of a similarity is a similarity.

Further preserving properties for quotients are given in the next theorem.

Theorem 5.5. Let $\equiv$ be a congruence in a fuzzy structure *M*. Then for every positive universal fuzzy formula $<\alpha,\lambda>$,

$$M \models <\alpha,\lambda> \Rightarrow M/\!\equiv\ \models <\alpha, [\lambda]> \qquad (5.6)$$

Assume that for every $\lambda\in V$ the class $[\lambda]$ is closed with respect to the *inf* operator. Then, for every identity-free fuzzy formula $<\alpha,\lambda>$,

$$M \models <\alpha,\lambda> \quad \Leftrightarrow \quad M/\!\equiv\ \models <\alpha, [\lambda]> \qquad (5.7)$$

Proof. Implication (5.6) is a consequence of *ii*) of Proposition 5.1. Equivalence (5.7) follows from Theorem 5.2.

6. Properties preserved by products and ultraproducts

In this section we will examine the properties preserved by the products and the ultraproducts. At first we will examine the behavior of the *inf* operator with respect to the direct product and the ultraproduct of the family of valuation structures.

Lemma 6.1. If $(S_i, \wedge_i)_{i\in I}$ is a family of semilattices and $Z \subseteq \Pi_{i\in I} S_i$ be such that $Inf(pr_i(Z))$ exists for every $i\in I$. Then *Z* admits a greatest lower bound and

$$Inf(Z) = <Inf(pr_i(Z))>_{i\in I}. \qquad (6.1)$$

Moreover, if $\mathcal{U}$ is an ultrafilter in *I* and *Z* is a rectangle, i.e. $Z = \Pi_{i\in I} Z_i$ where Z_i is a subset of S_i for every $i\in I$, then $[Z] = \{[z] : z \in Z\}$ admits a greatest lower bound and

$$Inf([Z]) = [<Inf(Z_i)>_{i\in I}]. \qquad (6.2)$$

Proof. If $<z_i>_{i\in I}\in Z$ then, for every $i\in I$, $z_i\in pr_i(Z)$ and therefore $z_i\geq Inf(Z_i)$. This proves that $<Inf(pr_i(Z))>_{i\in I}$ is lower bound for *Z*. Let $m = <m_i>_{i\in I}$ be a lower bound for *Z* and let $i\in I$. Then for every $x\in pr_i(Z)$ there is $z = <z_i>_{i\in I}$ in *Z* such that $z_i = x$. Since $m\leq z$, it is $m_i\leq z_i = x$. Then $m_i \leq Inf(pr_i(Z)))$ and therefore $m \leq <Inf(pr_i(Z))>_{i\in I}$.

To prove the second part of the proposition observe that it is evident that $[<Inf(Z_i)>_{i\in I}]$ is an lower bound of $[Z]$. Let $[m] = [<m_i>_{i\in I}]$ be a lower bound for $[Z]$ and assume that $[m]\wedge[<Inf\ X_i>_{i\in I}] = [m]$ is false, i.e. $\{i\in I : (Inf\ X_i)\wedge m_i \neq m_i\} \in \mathcal{U}$. Let $z = <z_i>_{i\in I}$ be a family in $\Pi_{i\in I} Z_i$ such that z_i satisfies the condition $x\wedge m_i \neq m_i$ if such a condition is satisfied by some element in X_i and such that z_i is any element in Z_i otherwise. Then since $(Inf\ Z_i)\wedge m_i \neq m_i$ entails the existenxe of $x\in Z_i$ such that $x\wedge m_i \neq m_i$, $\{i\in I : z_i\wedge m_i \neq m_i\} \supseteq \{i\in I : (Inf\ X_i)\wedge m_i \neq m_i\}$ and therefore $\{i\in I : z_i\wedge m_i \neq m_i\} \in \mathcal{U}$. This proves that $[z]$ is an element in $[Z]$ such that $[z]\wedge[m] \neq [m]$, in spite of the fact that $[m]$ is a lower bound for $[Z]$.

In accordance with such a lemma, if $(M_i)_{i\in I}$ is a family of fuzzy models such that all the valuation structures $\boldsymbol{VAL}(M_i)$ are complete, then the valuation structure

$\Pi_{i\in I}\,\boldsymbol{VAL}(M_i)$ of the product $\Pi_{i\in I}\ M_i$ is complete, too. This is not true in the case of an ultraproduct. As an example if all the valuation structures coincides with a valuation structure defined in the complete lattice [0,1], then the valuation structure in the ultraproduct is defined in the non-standard interval $[0,1]^*$ and such an interval is not complete.

Theorem 6.2. Let $(M_i)_{i\in I}$ be a family of safe fuzzy models and $M = \Pi_{i\in I} M_i$ its direct product. Then, M is safe, and for every formula α,

$$val(M, \alpha, f_1,\dots,f_n) = <val(M_i, \alpha, f_1(i),\dots, f_n(i))>_{i\in I} \tag{6.3}$$

for $f_1,\dots,f_n \in \Pi_{i\in I} D_i$. Consequently, $V(\alpha) = \Pi_{i\in I} V_i(\alpha)$ where $V(\alpha)$ and $V_i(\alpha)$ are the range of α in M and in M_i respectively. In particular, for every closed formula α,

$$val(M, \alpha) = <val(M_i, \alpha)>_{i\in I} \tag{6.4}$$

Proof. We operate by induction on the complexity of α.

Let $\alpha = \underline{r}(t_1,\dots,t_m)$, then

$$\begin{aligned} val(M, \alpha, f_1,\dots,f_n) &= I(\underline{r})(I(t_1)(f_1,\dots,f_n),\dots I(t_m)(f_1,\dots,f_n)) \\ &= <I_i(\underline{r})(I_i(t_1)(f_1(i),\dots,f_n(i)),\dots,I_i(t_m)(f_1(i),\dots,f_n(i)))>_{i\in I} \\ &= <val(M_i, \alpha, f_1(i),\dots,f_n(i))>_{i\in I}\,. \end{aligned}$$

Assume that (6.3) is satisfied by $\alpha_1,\dots,\alpha_t$. We have to prove that $\underline{c}(\alpha_1,\dots,\alpha_t)$ is valued in M and that such a formula satisfies (6.3). Indeed,

$$\begin{aligned} & val(M, \underline{c}(\alpha_1,\dots,\alpha_t), f_1,\dots,f_n) \\ &\quad = I(\underline{c})(val(M,\alpha_1, f_1,\dots,f_n),\dots,val(M,\alpha_t, f_1,\dots,f_n)) \\ &\quad = I(\underline{c})(<val(M_i, \alpha_1, f_1(i),\dots,f_n(i)>_{i\in I}\,,\dots,<val(M_i, \alpha_t, f_1(i),\dots,f_n(i)>_{i\in I}) \\ &\quad = <val(M_i, \underline{c}(\alpha_1,\dots,\alpha_t), f_1(i),\dots,f_n(i))>_{i\in I}\,. \end{aligned}$$

Now suppose that α is valued and that (6.3) is true for α. Then by (6.1),

$$
\begin{aligned}
val(M, \forall x_i\alpha, f_1,\dots,f_n) &= Inf\{val(M, \alpha, f, f_2,\dots,f_n) : f\in D\} \\
&= Inf\{<val(M_i, \alpha, f(i),\dots,f_n(i))>_{i\in I} : f\in D\} \\
&= <Inf\{val(M_i, \alpha, f(i),\dots,f_n(i)) : f(i)\in D_i\}>_{i\in I} \\
&= <val(M_i, \forall x_i\alpha, f_1(i),\dots,f_n(i))>_{i\in I}
\end{aligned}
$$

Note. (6.2) looks to be in contrast with the fact the in classical model theory only particular first order properties are preserved by the direct products. The contrast is only apparent since in the approach proposed in this thesis the product M of a family $(M_i)_{i\in I}$ of normal crisp models is not a crisp normal model and therefore it is not the usual product. As a matter of fact, the usual product is the 1-cut M_1 of M. Then, in accordance with Theorem 1.6, only in the case α is a positive clause we can claim that

$$M_i \vDash \alpha \text{ for every } i\in I \;\Leftrightarrow\; M \vDash \alpha \Rightarrow M_1 \vDash \alpha. \qquad (6.5)$$

Theorem 6.3. Let $(M_i)_{i\in I}$ be a family of safe fuzzy models and $\mathcal{U}$ be an ultrafilter in $P(I)$. Then the ultraproduct M^u of $(M_i)_{i\in I}$ modulo $\mathcal{U}$ is safe and

$$val(M^u, \alpha, [f_1],\dots,[f_n]) = [<val(M_i, \alpha, f_1(i),\dots, f_n(i))>_{i\in I}] \qquad (6.6)$$

for every formula α and $[f_1],\dots,[f_n]\in D$.

Proof. We prove (6.6) by induction on the complexity of α.

Let $\alpha = \underline{r}(t_1,\dots,t_m)$, then

$$
\begin{aligned}
val(M^u, \alpha, [f_1],\dots,[f_n]) &= I(\underline{r})(I(t_1)([f_1],\dots,[f_n]),\dots I(t_m)([f_1],\dots,[f_n])) \\
&= [<I_i(\underline{r})(I_i(t_1)(f_1(i),\dots,f_n{}^i(i),\dots I_i(t_m)(f_1(i),\dots,f_n(i)))>_{i\in I}] \\
&= [<val(M_i, \alpha, f_1(i),\dots, f_n(i))>_{i\in I}].
\end{aligned}
$$

Assume that $\alpha_1,...,\alpha_n$ are valued and that (6.6) is satisfied by $\alpha_1,\dots,\alpha_t$. Then, given a logical connective $\underline{o}$, we have to prove that $\underline{o}(\alpha_1,\dots,\alpha_t)$ is valued and that it satisfies (6.6). Indeed,

$$
\begin{aligned}
&val(M^u, \underline{o}(\alpha_1,\dots,\alpha_t), [f_1],\dots,[f_n]) \\
&\quad= I(\underline{o})(val(M^u,\alpha_1, [f_1],\dots,[f_n]),\dots,val(M^u,\alpha_t, [f_1],\dots,[f_n])) \\
&\quad= I(\underline{o})([<val(M_i, \alpha_1, f_1(i),\dots, f_n(i))>_{i\in I}],\dots,[<val(M_i, \alpha_t, f_1(i),\dots, f_n(i))>_{i\in I}]) \\
&\quad= [<I_i(\underline{o})(val(M_i, \alpha_1, f_1(i),\dots, f_n(i)),\dots,val(M_i, \alpha_t, f_1(i),\dots, f_n(i))>_{i\in I}]) \\
&\quad= [<val(M_i, \underline{o}(\alpha_1,\dots,\alpha_t), f_1(i),\dots, f_n(i)>_{i\in I}].
\end{aligned}
$$

Now suppose that α is valued and α satisfies (6.6). Then

$$val(M^u, \forall x_1\alpha, [f_1],\dots,[f_n]) = Inf\{val(M^u, \alpha, [f], [f_2],\dots,[f_n]) : f\in D\}$$

$$= Inf\{[<val(M_i, \alpha, f(i),\dots, f_n(i))>_{i\in I}] : f\in D\}.$$

In turn, if we set $V_i(\alpha) = \{val(M_i, \alpha, d,\dots, f_n(i)) : d\in D_i\}$, then, by Lemma 6.1,

$$Inf\{[<val(M_i, \alpha, f(i),\dots, f_n(i))>_{i\in I}] : f\in D\}$$

$$= Inf([\Pi_{i\in I} V_i(\alpha)]) = [<Inf(V_i(\alpha))>_{i\in I}]$$

$$= [<Inf\{val(M_i, \alpha, f(i),\dots,f_n(i)) : f(i)\in D_i\}>_{i\in I}]$$

$$= [< val(M_i, \forall x_1\alpha, f_1(i),\dots,f_n(i)) >_{i\in I}].$$

7. Modifying the valuation-scale of the predicates

In this section we will analyze the question of the properties preserved after a “deformation” of a fuzzy model. More in particular, after a modification of the valuation part of such a structure. In Section 1 we early considered such a question after the drastic modification obtained by “cutting” a fuzzy structure at a given level.

Definition 7.1. Let $M = (D,V,I)$ be a fuzzy interpretation, $\mathbf{V'} = (V', I')$ be a valuation structure and $k : V\to V'$ an order-preserving map such that $k(1) = 1$. Then we call *k-deformation* of (D,V,I) the interpretation $M_k = (D, V',I_k)$ defined by setting: $I_k(\underline{\lambda}) = I'(\underline{\lambda})$, $I_k(\underline{o}) = I'(\underline{o})$, $I_k(\underline{c}) = I(\underline{c})$ and $I_k(\underline{h}) = I(\underline{h})$ for every constant $\underline{c}$ and operation symbol $\underline{h}$ and

$$I_k(\underline{r})(d_1,...,d_n) = k(I(\underline{r})(d_1,...,d_n)) \qquad (7.1)$$

for every n-ary relation symbol $\underline{r}$ (different from =) and $d_1,...,d_n$ in D.

Such a definition extends Definition 1.3, obviously. The idea is that we can modify the valuation of the predicates in an uniform way. The question we are interested in is to individuate the properties of M inherited by M_k. A first immediate result is the following one.

Proposition 7.2. Let M_k be the deformation of M by k. Then, for every atomic formula α,

$$val(M_k, \alpha, d_1,...,d_n) = k(val(M, \alpha, d_1,...,d_n)). \qquad (7.2)$$

Proof. Assume α equal to $\underline{r}(t_1,...,t_p)$, then

$$val(M_k, \underline{r}(t_1,...,t_p), d_1,...,d_n) = I_k(\underline{r})(I_k(t_1)(d_1,...,d_n),..., I_k(t_p)(d_1,...,d_n))$$

$$= I_k(\underline{r})(I(t_1)(d_1,...,d_n),..., I(t_p)(d_1,...,d_n))$$

$$= k(I(\underline{r})(I(t_1)(d_1,...,d_n),..., I(t_p)(d_1,...,d_n))$$

$$= k(val(M, \underline{r}(t_1,...,t_p), d_1,...,d_n)).$$

To obtain more interesting results we have to assume that k is an homomorphism. In such a case the following proposition holds true.

Proposition 7.3. Let $M = (D,V,I)$ be a fuzzy structure and k be a homomorphism from the valuation structure ***V*** into another valuation structure ***V'***. Then k defines a valuation homomorphism from M to M_k and therefore:

i) for every universal fuzzy formula $<\alpha,\lambda>$ which is positive with respect to ***V'***

$$M \models <\alpha,\lambda> \Rightarrow M_k \models <\alpha, k(\lambda)> \qquad (7.3)$$

ii) assume that k is an isomorphism, then for every fuzzy formula $<\alpha,\lambda>$,

$$M \models <\alpha,\lambda> \Leftrightarrow M_k \models <\alpha, k(\lambda)> \qquad (7.4)$$

iii) for every universal fuzzy formula $<\alpha,\lambda>$ which is either identity-free or positive with respect to ***V'***

$$M \models \alpha \Rightarrow M_k \models \alpha \qquad (7.5)$$

Proof. Claim *i*) is a consequence of Proposition 4.1. Claim *ii*) follows from Theorem 4.3. Claim *iii*) is a consequence of Proposition 4.4.

Observe that in Proposition 7.3 it is required that all the operations in the considered valuation structure are preserved by k. In the case of fuzzy clauses more simple conditions are sufficient.

Proposition 7.4. Let ***V*** be a residuated lattice, $M = (D,V,I)$ be a fuzzy structure and let k be an order-preserving map such that $k(\lambda\otimes\mu) \geq k(\lambda)\otimes k(\mu)$ for every $\lambda, \mu \in V$. Then for every positive fuzzy clause $<\alpha_1\underline{\wedge}^{*}\ldots\underline{\wedge}^{*}\alpha_t\underline{\rightarrow}\alpha,\lambda>$

$$M \models <\alpha_1\underline{\wedge}^{*}\ldots\underline{\wedge}^{*}\alpha_t\underline{\rightarrow}\alpha,\lambda> \Rightarrow M_k \models <\alpha_1\underline{\wedge}^{*}\ldots\underline{\wedge}^{*}\alpha_t\rightarrow\alpha, k(\lambda)> \qquad (7.6)$$

Consequently, if M is a model of a fuzzy program p, then M_k is a model of the fuzzy program $k\circ p$.

Assume that k is injective, $\wedge$-preserving and such that $k(\lambda\otimes\mu) \leq k(\lambda)\otimes k(\mu)$ for every $\lambda, \mu \in V$. Then,

$$M \models <\alpha_1\underline{\wedge}^{*}\ldots\underline{\wedge}^{*}\alpha_t\rightarrow\alpha,\lambda> <= M_k \models <\alpha_1\underline{\wedge}^{*}\ldots\underline{\wedge}^{*}\alpha_t\rightarrow\alpha, k(\lambda)> \qquad (7.7)$$

and therefore if M_k is a model of the fuzzy program $k \circ p$ then M is a model of a fuzzy program p.

Proof. By Proposition 1.2 in Chapter 3, if M satisfies $<\alpha_1\underline{\land}^*\ldots\underline{\land}^*\alpha_t\underline{\rightarrow}\alpha,\lambda>$ then

$$val(M,\ \alpha_1,\ d_1,...,d_n)\otimes...\otimes val(M,\ \alpha_t,\ d_1,...,d_n)\otimes\lambda \leq val(M,\ \alpha,\ d_1,...,d_n)$$

for every $d_1,\ldots,d_n$ in D. Consequently

$$\begin{aligned} &k(val(M,\alpha_1,\ d_1,...,d_n))\otimes...\otimes k(val(M,\ \alpha_t,\ d_1,...,d_n))\otimes k(\lambda) \\ &\qquad \leq k(val(M,\alpha_1,\ d_1,...,d_n)\otimes...\otimes val(M,\ \alpha_t,\ d_1,...,d_n))\otimes k(\lambda) \\ &\qquad \leq k(val(M,\ \alpha,\ d_1,...,d_n)) \end{aligned}$$

and therefore, by (7.2),

$$val(M_k,\ \alpha_1,\ d_1,\ldots,d_n)\otimes...\otimes val(M_k,\ \alpha_t,\ d_1,...,d_n))\otimes k(\lambda) \leq val(M_k,\ \alpha,\ d_1,...,d_n)).$$

In turn, this means that M_k satisfies $<\alpha_1\underline{\land}^*\ldots\underline{\land}^*\alpha_t \rightarrow\alpha,\ k(\lambda)>$

To prove the second part, assume that $M_k \leq <\alpha_1\underline{\land}^*\ldots\underline{\land}^*\alpha_t \rightarrow\alpha,\ k(\lambda)>$, i.e.

$$k(val(M,\alpha_1,\ d_1,...,d_n))\otimes...\otimes k(val(M,\ \alpha_t,\ d_1,...,d_n))\otimes k(\lambda) \leq k(val(M,\ \alpha,\ d_1,...,d_n))$$

Then

$$\begin{aligned} &k(val(M,\alpha_1,\ d_1,...,d_n))\otimes...\otimes val(M,\ \alpha_t,\ d_1,...,d_n))\otimes\lambda) \\ &\qquad \leq k(val(M,\alpha_1,\ d_1,...,d_n))\otimes...\otimes k(val(M,\ \alpha_t,\ d_1,...,d_n))\otimes k(\lambda) \\ &\qquad \leq k(val(M,\ \alpha,\ d_1,...,d_n)) \end{aligned}$$

Consequently, since k is injective and $\land$-preserving,

$$val(M,\alpha_1,\ d_1,...,d_n))\otimes...\otimes val(M,\ \alpha_t,\ d_1,...,d_n))\otimes\lambda \leq val(M,\ \alpha,\ d_1,...,d_n)$$

and therefore $M \vDash <\alpha_1\underline{\land}^*\ldots\underline{\land}^*\alpha_t \rightarrow\alpha,\lambda>$.

Example 1. Let $\otimes$ be a triangular norm, and define the $\otimes$-n-power $\lambda^{(n)}$ by setting $\lambda^{(1)} = \lambda$ and $\lambda^{(n)} = \lambda^{(n-1)}\otimes\lambda$. Then, trivially, by setting $k_n(\lambda) = \lambda^{(n)}$, we obtain a map such that $k_n(\lambda\otimes\mu) = k_n(\lambda)\otimes k_n(\mu)$ and which is $\land$-preserving. In the case $\otimes$ is the usual product, such a map is injective. If we set $k(\lambda) = \lambda^n$ and $\otimes$ is the product of Lukasiewicz, then it is possible to prove that $k(\lambda\otimes\mu) \geq k(\lambda)\otimes k(\mu)$. In accordance, if (S,e) is a $\otimes$-similarity, then we obtain a similarity $e_{(n)}$ by setting $e_{(n)}(x,y) = e(x,y)^{(n)}$. In the case $\otimes$ is the Lukasiewicz product another similarity is obtained by setting $e_n(x,y) = e(x,y)^n$.

Example 2. To show an example of property which is not preserved, consider the valuation structure $(\{0, ½, 1\}, \land, \rightarrow, \neg, 0, 1)$ where $\land$ is the minimum, $\rightarrow$ is the corresponding implication and $\neg$ is the 1-ary operation such that $\neg(x) = 1\text{-}x$. Also, consider the fuzzy subgroup M_4 defined in the additive group $(Z_4, +, ^{-1}, 1)$ of integers modulo 4 by the fuzzy subset $s : Z_4 \rightarrow \{0, ½, 1\}$ such that $s(0) = 1$, $s(1) = 0$, $s(2) = ½$, $s(3) = 0$. Finally, denote by $k : \{0, ½, 1\} \rightarrow \{0, ½, 1\}$ the function such that $k(0) = 0$; $k(1/2) = 1$; $k(1) = 1$. Then, the deformation of M_4 by k is a fuzzy subgroup which is not a model of $\exists x(\underline{s}(x) \leftrightarrow\underline{\neg}\ \underline{s}(x))$. This is in accordance with the fact that *f* is not a homomorphism since it is not compatible with the interpretation of the negation.

8. Further results on the deformation

Another way to modify a model is to consider a quotient of the valuation part of a structure and by applying the results in Section 5. Notice that it is possible to obtain the following very simple characterization of the congruences in the structure ([0,1], ∧, →, 0, 1).

Proposition 8.1. Consider the valuation structure ([0,1], ∧, →, 0, 1) where ∧ is the operation of minimum and → the related residuum. Then we can identify the congruences in such a structure with the partitions in which a class is an interval containing 1 and the remaining classes are singletons.

Example 1. For instance consider the fuzzy subgroup M_{16} defined in Example 1 of Section 2 and define in ([0,1], ∧, →,0,1) the congruence ≡ generated by the pair (0.3, 1). Then ≡ is the congruence whose classes are the interval [0.3, 1] together with the singletons {*x*} with $x \notin$ [0.3,1]. The related quotient is

Z	0	1	2	3	4	5	6	7
G(*z*)	[0.3,1]	{0.1}	{0.2}	{0.1}	[0.3,1]	{0.1}	{0.2}	{0.1}
Z	8	9	10	11	12	13	14	15
G(*z*)	[0.3,1]	{0.1}	{0.2}	{0.1}	[0.3,1]	{0.1}	{0.2}	{0.1}

Equivalently, if we denote by 0.3 the whole class [0.3] and we identify a singleton {*x*} with *x*, then we can identify the quotient of the valuation structure with ([0, 0.3], ∧, →, 0, 0.3) and therefore to represent the quotient as follows:

z	0	1	2	3	4	5	6	7
g(*z*)	0.3	0.1	0.2	0.1	0.3	0.1	0.2	0.1
z	8	9	10	11	12	13	14	15
g(*z*)	0.3	0.1	0.2	0.1	0.3	0.1	0.2	0.1

Up to now we considered homomorphisms and, in particular, endomorphisms. Unfortunately there are basic valuation structures which are *rigid*, i.e. in which the only endomorphism is the identity map. Consider for example the interval [0,1] equipped with the Lukasiewicz norm ⊗ and the related residuum and negation. Then if k is an endomorphism, all the rational numbers are fixed points. Indeed, taking in account of the fact that k preserves also the operation ⊕, given $m \in N$-{0}, since

$$1 = k(1) = k(m/m) = k(1/m \oplus \ldots \oplus 1/m) = k(1/m) \oplus \ldots \oplus k(1/m) = m \cdot k(1/m)$$

we have that $k(1/m) = 1/m$. Also, for every n,

$$k(n/m) = k(1/m \oplus \ldots \oplus 1/m) = k(1/m) \oplus \ldots \oplus k(1/m) = n/m$$

Since k is order-preserving, this entails that all the real numbers in [0,1] are fixed points. Thus there is no non-trivial endomorphism in the Lukasievicz valuation structure.

These considerations suggest a different strategy in which we admit also a modification of the operations in a valuation structure.

Definition 8.2. Let $\boldsymbol{V} = (V,I)$ be a valuation structure and $k : V \rightarrow V$ be an order-preserving one-to-one map. Then we denote by $\boldsymbol{V}_k$ the valuation structure (V,I_k) whose domain is V and in which an n-ary logical connective $\underline{c}$ is interpreted by setting, for every $\lambda_1,\ldots,\lambda_n$ in V,

$$I_k(\underline{c})(\lambda_1,\ldots,\lambda_n) = k(I(\underline{c})\,(k^{-1}(\lambda_1),\ldots,\, k^{-1}(\lambda_n))) \qquad .$$

Notice that, in account of the fact that k preserves the meet operator, $I_k(\underline{\wedge}) = I(\underline{\wedge})$. As an example, consider a valuation structure as ([0,1], ∧, ⊗, 0, 1) where ⊗ = $I(\underline{\otimes})$ is the interpretation of a binary logical connective $\underline{\otimes}$ and assume that $k : [0,1] \rightarrow [0,1]$ is an order-preserving one-to-one map. Then, k is a ∧-automorphism and the operation $\otimes_k = I_k(\underline{\otimes})$ is define by

$$x \otimes_k y = k(k^{-1}(x) \otimes k^{-1}(y)).$$

The proof of the following proposition is trivial

Proposition 8.3. Let $\boldsymbol{V} = (V,I)$ be a valuation structure and $k : V \rightarrow V$ be an order-preserving one-to-one map. Then k is an isomorphism between $\boldsymbol{V}$ and $\boldsymbol{V}_k$..

Observe that from such a proposition it follows if ⊗ is a triangular norm then $\otimes_k$ is a triangular norm, too.

Definition 8.4. Let $M = (D,V,I)$ be a fuzzy structure and let $k : V \rightarrow V$ be an order-preserving one-to-one map. Then we call *total k-deformation* of (D,V,I) the fuzzy model M_k which is the k-deformation by the isomorphism k from $\boldsymbol{V}$ to $\boldsymbol{V}_k$..

The proof of the following proposition is obvious.

Theorem 8.5. Let $M = (D,V,I)$ be a fuzzy structure and $k : V \to V$ be an order-preserving, one-to-one map. Then, for every $\lambda \in V$ and every formula α,

$$M \vDash <\alpha,\lambda> \Leftrightarrow M_k \vDash <\alpha, k(\lambda)>.$$

As an example, if we consider a ⊗-similarity $e: S \times S \to [0,1]$ and $k : [0,1] \to [0,1]$ an one-to-one order preserving map, then by setting $e_k(x,y) = k(e(x,y))$ we obtain a $\otimes_k$-similarity. We can verify directly such a fact since

$$e_k(x,x) = k(e(x,x)) = k(1) = 1 \quad \text{and} \quad e_k(x,y) = k(e(x,y)) = k(e(y,x)) = e_k(y,x).$$

Moreover, since $e(x,y))\otimes e(y,z) \leq e(x,z)$,

$$k(e(x,y)) \otimes e(y,z)) \leq k(e(x,z))$$

and therefore

$$k(e(x,y)) \otimes_k k(e(y,z)) = k(k^{-1}(k(e(x,y)) \otimes k^{-1}(k(e(y,z)))$$

$$= k(e(x,y) \otimes e(y,z)) \leq k(e(x,z)).$$

This proves that

$$e_k(x,y) \otimes_k e_k(y,z) \leq e_k(x,z)$$

Likewise, we have that the flexible deformation of a fuzzy subgroup with respect to the valuation structure $\boldsymbol{V} = ([0,1], \wedge, \otimes, 0, 1)$ is a fuzzy subgroup with respect to the valuation structure $\boldsymbol{V}_k = ([0,1], \wedge, \otimes_k, 0, 1)$.

REFERENCES

[1] Alsinet T., Godo L., Adding similarity-based reasoning capabilities to a Horn fragment of possibilistic logic with fuzzy constants, *Fuzzy Sets and Systems*, 144 (2004) 43-65.

[2] Biacino L., Gerla G., Recursively enumerable *L*-sets, *Zeitschr. f. math. Logik und Grundlagend Math.*, 33 (1987) 107-113.

[3] Biacino L., Gerla G., Logics with approximate premises, *International J. of Intell. Syst.*, 13 (1998) 1-10.

[4] Biacino L., Gerla G., Closure Operators in Fuzzy Set Theory, in *Fuzzy Sets in Approximate Reasoning and Information Systems*, Edited by J. C. Bezdek, D. Dubois, H. Prade, Kluwer Academic Publishers, Dordrecht, (1999) 243-278.

[5] Biacino L., Gerla G., Ying M., Approximate reasoning based on similarity, *Math. Log. Quart.*, 46 (2000) 77-86.

[6] Cignoli R., D'Ottaviano I.M.L. and Mundici D., Algebraic Foundations of many-valued Reasoning, *Trends in Logic*, 7, Kluwer Academic Publishers, Dordrecht, (2000).

[7] Costantini S., Meta-reasoning: a survey, *Proc. Computational Logic: Logic Programming and Beyond*, Essays in Honour of Robert A. Kowalski, A. C. Kakas e F. Sadri (Eds.), Springer-Verlag Lecture Notes in Artificial Intelligence 2407, (2002)

[8] Deschrijver G., Arieli O., Cornelis C., Kerre E.: A Bilattice-Based Framework for Handling Graded Truth and Imprecision. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 15(1) (2007) 13-41.

[9] Ebrahim, R., Fuzzy logic programming, *Fuzzy Sets and Systems*, 117 (2001) 215-230.

[10] Fitting M., Bilattices and the semantics of logic programming. *Journal of Logic Programming*, 11 (1991) 91–116.

[11] 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) 51-68.

[12] Formato F., Gerla G., Sessa M., Similarity-based unification, *Fundamenta Informaticae*, 41, (2000) 393-414.

[13] Genito D., Gerla G., Vignes A., Meta-logic Programming for synonymy logic, *Soft Computing*, 14 (2010) 299-311.

[14] Genito D., Gerla G., An attempt to connect bilattice theory with fuzzy logic, submitted to *Archive for Mathematical Logic*.

[15] Genito D., Gerla G., Vignes A., Cuts and deformations for fuzzy structures, submitted to *Fuzzy Set and Systems*.

[16] Gerla G., The category of the fuzzy models and Lowenheim-Skolem Theorem, in *Mathematics of Fuzzy Systems*, Di Nola and Ventre (eds), Verlag TUV Rheinland, Germany (1986) 121-141.

[17] Gerla G., Fuzzy Logic: Mathematical Tools for Approximate Reasoning, *Kluwer Academic Publishers*, (2001).

[18] Gerla G., Fuzzy Logic Programming and fuzzy control, *Studia Logica*, 79 (2005) 231-254

[19] Gerla G., Effectiveness and Multivalued Logics, *Journal of Symbolic Logic*, 71 (2006) 137-162.

[20] Ginsberg M., Multivalued logics: a uniform approach to inference in artificial intelligence, *Computer Intelligence*, 4, (1988) 265-316.

[21] Gottwald S., Fuzzy Sets and Fuzzy Logic, *Vieweg*, Wiesbaden, (1993).

[22] Hájek P., Metamathematics of Fuzzy Logic, *Kluwer Academic Publishers*, Dordrecht, (1998).

[23] Hill P. M., Gallagher J., *Meta-Programming in Logic Programming*. In Artificial Intelligence and Logic Programming. Oxford University Press, (1994).

[24] Lloyd J.W., Foundations of Logic Programming, *Springer-Verlag*, Berlin, (1987) .

[25] Medina J., Ojeda-Aciego M., Vojtáš P., Similarity-based unification: a multi-adjoint approach, *Fuzzy Sets and Systems*, 146 (2004) 43-62.

[26] Nikolov N. G., Deductive properties of intuitionistic fuzzy logic, *Fifth Int. Conf. on IFSs* NIFS 7, 3 (2001) 43-48.

[27] Novak V., Fuzzy sets and their applications, Bristol: Adam Hilger, (1989).

[28] Pavelka J., On fuzzy logic I: Many-valued rules of inference, *Zeitschr. f. math. Logik und Grundlagen d. Math.*, 25 (1979) 45-52.

[29] Pilitowska A., Interval bilattices and some other simple bilattices, In *Revised Papers From the 6th international Conference and 1st Workshop of COST Action 274 TARSKI on Relational Methods in Computer Science* (October 16 - 21, 2001). H. C. Swart, Ed. Lecture Notes In Computer Science, Springer-Verlag, London, 2561 (2002) 190-196.

[30] Valverde L., On the Structure of F-Indistinguishability Operators. *Fuzzy Sets and Systems*, 17 (1985) 313-328.

[31] Vojtáš P., Fuzzy logic programming, *Fuzzy Sets and Systems*, 124 (2001) 361-370.

[32] Wiedermann J., Fuzzy Turing Machines revised, *Inform. and Computing*, 21 (2002) 1-13.

[33] Ying M., A logic for approximate reasoning, *The Journal of Symbolic Logic*, 59 (1994) 830-837.

[34] Zadeh L.A., Fuzzy Sets, *Information and Control* 8 (1965) 338-353.