Multi-valued Logics, Effectiveness and Domains

Giangiacomo Gerla

Department of Mathematics and Computer Science,
University of Salerno
Via Ponte don Melillo 84084,
Fisciano (SA) ITALY gerla@unisa.it

Key words: Multi-valued logic, Fuzzy logic, Computability, Domain theory, Fuzzy grammar, Fuzzy Turing Machine, Church Thesis.

Abstract. Effective domain theory is applied to fuzzy logic to give suitable notions of semi-decidable and decidable L-subset. The connection with the notions of fuzzy Turing machines and fuzzy grammar given in literature is also investigated. This shows the inadequateness of these definitions and the difficulties in formulating an analogue of Church Thesis for fuzzy logic.

1 Introduction

Fuzzy logic is a promising chapter of multi-valued logic whose basic ideas have been formulated by L. A. Zadeh, J. A. Goguen, J. Pavelka and others (see, for example, [5], [17] and [12]) and successively investigated by several authors (see, for example, [7], [11], [6], [4]). The aim of such a logic is to formalize the *"approximate reasoning"* we use in everyday life where vague notions, such as *big*, *slow*, *near*, are constantly involved. This leads to define a deduction operator associating every fuzzy subset of axioms with the related fuzzy subset of logical consequences. Now it is evident that a basic task for fuzzy logic is to exhibit the effectiveness of its deduction apparatus. In particular, it is important to prove that the fuzzy subset of consequences of a "decidable" fuzzy subset of axioms is "effectively enumerable". To do this we have to give adequate definitions of "effective enumerability" and "decidability" for fuzzy subsets.

On the other hand, in my opinion the phenomenon of the vagueness leads to assume that the set of truth values is a continuum. More precisely, density is suggested by the existence of intermediate values. To give an example, assume that the atomic formula $Big(a)$ is evaluated λ, that $Big(b)$ is evaluated μ and $\lambda < \mu$. Then we cannot exclude the existence of an object d such that $Big(d)$ has a truth value between λ and μ. Also, completeness is suggested by the fact that the quantifiers are interpreted by the least upper bound and the greatest lower bound operators. Moreover, the existence of the least upper bounds is necessary to fuse the different valuations given by different proofs of a given formula (see [12]).

Once we accept the hypothesis that the set of truth values is a continuum, the notion of effectiveness has to be based on endless effective approximation algorithms (as in recursive analysis) and not on algorithms converging in finite steps (as in recursive arithmetics). So, as proposed in [3], a natural framework to define the notion of effectiveness in multi-valued logic is the theory of effective domains (see also [2], [1]). Obviously, this is not the unique possible choice and it is possible to refer to the vast and interesting literature concerning a constructive approach to the continuum.

In this paper we compare the domain-based definition of semi-decidability for fuzzy subsets with the definitions given in literature based on the notions of fuzzy grammar and fuzzy Turing machine. This comparison proves that these definitions are not adequate and it shows the difficulties in formulating an analogue of Church Thesis for fuzzy logic. Also, it emphasizes an open question: to find adequate definitions of multi-valued Turing machine and multi-valued grammar.

2 Preliminaries: Effective lattices and semi-decidable elements

In this paper L always denotes a complete lattice with minimum 0 and maximum 1. Given $x, y \in L$, we say that x is *way below* y and we write $x \ll y$ provided that, for every nonempty upward directed subset A of L

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

Definition 1. *A based continuous lattice, in brief a based lattice, is a structure* $(L, \leq, B)$ *where* L *is a complete lattice and* B*, the basis, is a subset of* L *containing 0, closed with respect to* $\vee$ *and* $\wedge$ *and such that, for every* $x \in L$*,*

$$x = \sup(\{b \in B : b \ll x\}). \quad (1)$$

We have that $x \ll y$ entails $x \leq y$. If L is a finite chain and we set $B = L$, then $(L, \leq, B)$ is a based lattice such that

$$x \ll y \Leftrightarrow x \leq y.$$

If L is a complete chain and B a dense subset of L, then $(L, \leq, B)$ is a based lattice such that

$$x \ll y \Leftrightarrow \text{ either } x = 0 \text{ or } x < y.$$

Definition 2. *An effective continuous lattice (see [14]), in brief an effective lattice, is a based lattice* $(L, \leq, B)$ *with an enumeration* $(b_n)_{n \in N}$ *of* B *such that*

- *the relation* $\{(n, m) \in N^2 : b_n \ll b_m\}$ *is recursively enumerable*
- *two recursive maps join*: $N \times N \to N$, *meet* : $N \times N \to N$ *exist such that*

$$b_n \vee b_m = b_{join(n,m)} \ , \ b_n \wedge b_m = b_{meet(n,m)}.$$

In brief, an effective lattice is a based lattice such that in B the relation $\ll$ is recursively enumerable and $\vee$, $\wedge$ are computable operations.

It is evident that every finite chain L is an effective lattice with respect to $B = L$. The interval $U = [0, 1]$ is an effective lattice whose basis is the set U_Q of rational numbers in U.

Definition 3. *We say that an element x in an effective lattice $(L, \leq, B)$ is semi-decidable if the cut $\{n \in N : b_n \ll x\}$ is recursively enumerable.*

In particular, every $b \in B$ is semi-decidable and 1 is semi-decidable, too. In a finite lattice all the elements are semi-decidable. If $L = U$, then

$$x \text{ is semi-decidable} \Leftrightarrow \{r \in U_Q : r < x\} \text{ is recursively enumerable.}$$

Proposition 1. *Let $(L, \leq, B)$ be an effective lattice, then the following are equivalent:*

i) *x is semi-decidable,*
ii) *a recursive map f exists such that $(b_{f(n)})_{n \in N}$ is $\ll$-preserving and*

$$x = \sup_{n \in N} b_{f(n)}, \tag{2}$$

iii) *a recursive map f satisfying (2) exists such that $(b_{f(n)})_{n \in N}$ is order-preserving,*
iv) *a recursive map f satisfying (2) exists.*

3 Decidable elements

To define the notion of decidability we need to dualize some of the definitions in the previous sections. Given a lattice $(L, \leq)$, we denote by $(L, \leq_d)$ its dual. Any order-theoretical concept in $(L, \leq)$ is associated with its dual, i.e. the same concept interpreted in $(L, \leq_d)$. As an example, we say that y is *way above* x and we write $x \ll^d y$ in the case y is way below x in $(L, \leq_d)$. Then $x \ll^d y$ if, for every downward directed subset A of L,

$$x \geq \inf A \Rightarrow \text{ there exists } a \in A \text{ such thath } \; y \geq a.$$

Obviously $x \ll^d y$ entails $y \leq x$. If L is a finite chain,

$$x \ll^d y \Leftrightarrow y \leq x.$$

If L coincides with U, then

$$x \ll^d y \Leftrightarrow \text{either } y = 1 \text{ or } y < x.$$

Definition 4. *A structure $(L, \leq, B, \underline{B})$ is called an (effective) ab-lattice, provided that both the structures $(L, \leq, B)$ and $(L, \leq_d, \underline{B})$ are based (effective) continuous lattices. In such a case we say that $B = (b_n)_{n \in N}$ is the basis and $\underline{B} = (\underline{b}_n)_{n \in N}$ the dual basis of $(L, \leq_d, B, \underline{B})$.*

Obviously, in an ab-lattice L we have that, for every $x \in L$,

$$x = \sup\{b \in B : b \ll x\} = \inf\{\underline{b} \in \underline{B} : x \ll^d \underline{b}\},$$

i.e. it is possible to approximate x both from below and from above.

Definition 5. *Given an ab-lattice $(L, \leq, B, \underline{B})$, we say that x is decidable if x is semi-decidable both in $(L, \leq, B)$ and in $(L, \leq_d, \underline{B})$, i.e. if both the cuts*

$$\{n \in N : b_n \ll x\} \quad ; \quad \{n \in N : x \ll^d \underline{b}_n\}$$

are recursively enumerable.

Trivially, in all the ab-lattices both 0 and 1 are decidable. The proof of the following proposition is an immediate consequence of Proposition 1.

Proposition 2. *Given an element x of an effective ab-lattice, the following are equivalent:*
i) x is decidable
ii) two total recursive functions $h : N \to N$, $k : N \to N$ exist such that $(b_{h(n)})_{n\in N}$ is $\ll$-preserving, $(\underline{b}_{k(n)})_{n\in N}$ is $\ll^d$-reversing and

$$\sup_{n\in N} b_{h(n)} = x = \inf_{n\in N} \underline{b}_{k(n)} \qquad (3)$$

iii) two total recursive functions $h : N \to N$, $k : N \to N$ exist such that (3) is satisfied, $(b_{h(n)})_{n\in N}$ is order-preserving and $(\underline{b}_{k(n)})_{n\in N}$ is order-reversing
iv) a nested effectively computable sequence $([b_{h(n)}, \underline{b}_{k(n)}])_{n\in N}$ of intervals exists such that

$$\{x\} = \bigcap_{n\in N} [b_{h(n)}, \underline{b}_{k(n)}].$$

An easy way to obtain ab-lattices is by an involution in L.

Definition 6. *A structure $(L, \leq, -, B)$ is an effective lattice with an involution if $(L, \leq, B)$ is an effective lattice and $-$ is an involution such that $\{(n, m) \in N \times N : -b_n \ll -b_m\}$ is recursively enumerable.*

In the case $L = \{\lambda_0, ..., \lambda_n\}$ is a finite chain where $0 = \lambda_0 < ... < \lambda_n = 1$, there is a unique involution $\neg$ defined by setting $\neg(\lambda_i) = \lambda_{n-i}$. In the case L is the interval U, an involution $\neg$ is obtained by setting $\neg(\lambda) = 1 - \lambda$.

Since an involution is an isomorphism between L and its dual and since an isomorphism preserves the definable relations, we have that, for every $x \in L$:

$$x \ll^d y \quad \Leftrightarrow \quad -y \ll -x.$$

The proof of the following proposition is trivial.

Proposition 3. *Let $(L, \leq, B, -)$ be an effective lattice with an involution and set $\underline{B} = (\underline{b}_n)_{n\in N}$ where $\underline{b}_n = -b_n$. Then $(L, \leq, B, \underline{B})$ is an effective ab-lattice. Moreover,*

x is decidable $\Leftrightarrow$ both x and $-x$ are semi-decidable.

This proposition entails that a finite chain L is an effective ab-lattice in which $B = \underline{B} = L$ and in which all the elements are decidable. The interval U is an effective ab-lattice in which $B = \underline{B} = U_Q$. In such a case an element x is decidable provided that both the sections $\{r \in U_Q : r < x\}$ and $\{r \in U_Q : x > r\}$ are recursively enumerable, i.e. x is a recursive real number.

4 The effective lattice of the L-subsets of a given set

Let S be a nonempty set. Then we call *L-subset* of S every element in the direct power L^S. We denote by $\cup$ and $\cap$ the lattice operations in L^S and we call these operations *union* and *intersection*, respectively. Then the union and intersection operations are defined by setting, for every $s_1, s_2 \in L^S$ and $x \in S$,

$$(s_1 \cup s_2)(x) = s_1(x) \vee s_2(x) \quad ; \quad (s_1 \cap s_2)(x) = s_1(x) \wedge s_2(x).$$

In an analogous way we define the infinitary unions and intersections. If $L = U$ an $L-$subset is also called *fuzzy subsets* of S. In the case an involution $\neg : L \to L$ is defined in L, then we call *complement* the corresponding operation in L^S. Then, the complement of an L-subset s, is the L-subset $-s$ defined by setting $(-s)(x) = \neg s(x)$. The elements in L are interpreted as truth values in a multi-valued logic where 0 is interpreted as "true" and 1 as "false". An L-subset is interpreted as a generalized characteristic function to represent the extension of a vague predicate. So, for every $x \in S$, $s(x)$ is the *membership degree* of x to s. We call *crisp* an L-subset s such that $s(x) \in \{0, 1\}$ for every $x \in S$. Given $X \in P(S)$, the *characteristic function* of X is the map $c_X : S \to L$ defined by setting $c_X(x) = 1$ if $x \in X$ and $c_X(x) = 0$ otherwise. We can identify the classical subsets of S with the crisp L-subsets of S via the characteristic functions.

Given an L-subset s, we set $Supp(s) = \{x \in S : s(x) \neq 0\}$ and $Cosp(s) = \{x \in S : s(x) \neq 1\}$. We say that s is *finite* (*co-finite*) provided that $Supp(s)$ ($Cosp(s)$, respectively) is finite. We call finite also the empty set and co-finite the whole set S. The classes of finite and co-finite L-subsets of S are denoted by $Fin(L^S)$ and $Cof(L^S)$, respectively. Obviously, if a negation is defined in L, then an L-subset is finite if and only if its complement is co-finite.

In the following we assume that S admits a code. This enables us to identify S with the set of natural numbers and to prove the following theorems (see [3]).

Theorem 1. *Let $(L, \leq, B)$ be an effective lattice. Then the class L^S of L-subsets of S is an effective lattice admitting as a basis the class $Fin(B^S)$ of finite L-subsets of S with values in B. Also, for every s_1 and s_2 in L^S,*

$$s_1 \ll s_2 \Leftrightarrow s_1 \text{ is finite and } s_1(x) \ll s_2(x) \text{ for every } x \in S.$$

Observe that, by definition, an L-subset s is semi-decidable provided that

$$\{n \in N : b_n \ll s\} = \{n \in N : b_n(i) \ll s(i) \text{ for every } i \in Supp(b_n)\}.$$

is a recursively enumerable set. There are simple characterizations of the semi-decidable L-subsets.

Theorem 2. *Let $(L, \leq, B)$ be an effective continuous lattice and $s \in L^S$. Then the following are equivalent:*
i) s is semi-decidable
ii) a recursive function $h : S \times N \to B$ exists which is $\ll$-increasing with respect to n such that

$$s(x) = \sup_{n \in N} h(x, n)$$

iii) a recursive function $h : S \times N \to B$ exists which is increasing with respect to n such that

$$s(x) = \sup_{n \in N} h(x, n).$$

The following proposition enables us to define the notion of decidable $L-$subset.

Proposition 4. *Let $(L, \leq, B, \underline{B})$ be an effective ab-lattice. Then L^S is an effective ab-lattice with dual basis the class $Cof(B^S)$ of co-finite L-subsets of S with values in B. If $(L, \leq, B, -)$ is an effective lattice with an involution, then L^S is an effective lattice with the complement as an involution.*

Trivially, if L is an effective lattice with an involution, then
s is decidable $\Leftrightarrow$ both s and its complement $-s$ are semi-decidable.

5 The main cases

In this section we will consider two cases which are basic ones in fuzzy logic: the finite chains and the interval U. Observe that in these cases the proposed notions of semi-decidability and decidability for fuzzy subsets are in accordance with the ones given in [1] and [2].

Proposition 5. *Let L be a finite chain. Then the class L^S of L-subsets of S is an effective lattice with the complement as an involution and therefore it is an effective ab-lattice. Its basis is the class $Fin(L^S)$ of finite L-subsets of S, its dual basis is the class $Cof(L^S)$ of co-finite L-subsets of S. Also*
$s_1 \ll s_2 \Leftrightarrow s_1 \subseteq s_2$ *and* s_1 *is finite*
and
$s_1 \ll^d s_2 \Leftrightarrow s_1 \subseteq s_2$ *and* s_2 *is co-finite.*

In particular, the class $P(S)$ of subsets of S is an effective lattice with an involution whose basis is the class of finite subsets and whose dual basis is the class of co-finite subsets of S. Also $X_1 \ll X_2 \Leftrightarrow X_1 \subseteq X_2$ and X_1 is finite
and
$X_1 \ll^d X_2 \Leftrightarrow X_1 \subseteq X_2$ and X_2 is co-finite.
Moreover, the proposed notions of decidability and semi-decidability coincide with the classical ones.

Proposition 6. *Let L be a finite chain. Then an L-subset s is semi-decidable if and only if there is a recursive function $h : S \times N \to L$ increasing with respect to the second variable such that*

$$s(x) = \max_{n \in N} h(x, n).$$

Moreover, s is decidable if and only if s is a recursive function.

The following proposition shows that, in the case of a finite chain, the proposed definition of semi-decidability is the only possible extension of the classical one such that

- the constant L-subsets are semi-decidable
- the union of two semi-decidable L-subsets is semi-decidable
- the intersection of two semi-decidable L-subsets is semi-decidable.

To show this, given an L-subset s, we call *closed λ-cut of s* the subset $C(s, \lambda) = \{x \in S : s(x) \geq \lambda\}$ where $\lambda \in L$. The equation

$$s(x) = \bigcup_{\lambda \in L} \lambda \wedge C(s, \lambda)$$

shows that the lattice of the L-subsets is the lattice generated by the constant L-subsets and the crisp L-subsets.

Proposition 7. *Let L be a finite chain. Then, the following are equivalent:*
i) s is a semi-decidable L-subset
ii) all the cuts of s are recursively enumerable.
As a consequence, the lattice of the semi-decidable L-subsets is the lattice generated by the recursively enumerable subsets and the constant L-subsets.

In the case $L = U$ we can prove a proposition similar to Proposition 5.

Proposition 8. *The class of fuzzy subsets of S is an effective lattice with the complement as an involution. The basis is the class $Fin(U_Q{}^S)$ of finite fuzzy subsets of S with rational values. The dual basis is the class $Cof(U_Q{}^S)$ of co-finite fuzzy subsets of S with rational values. Moreover*

$$s_1 \ll s_2 \;\Leftrightarrow\; s_1 \text{ is finite and } s_1(x) < s_2(x) \text{ for every } x \in Supp(s_1).$$

$$s_1 \ll^d s_2 \;\Leftrightarrow\; s_2 \text{ is co-finite and } s_1(x) < s_2(x) \text{ for every } x \in Cosp(s_1).$$

Unfortunately, we cannot extend Proposition 7 to this case since a closed cut of a semi-decidable L-subset is not necessarily recursively enumerable. More precisely, we have the following proposition whose proof is an immediate consequence of a series of interesting results about the effectiveness in multi-valued logic (see for example [7] and [10]).

Theorem 3. *A subset of S is a closed cut of a semi-decidable fuzzy subset iff it belongs to the Σ_2-level of the arithmetical hierarchy.*

Observe that such a theorem gives an explanation of an apparent contradiction. In fact the scholars interested in multi-valued logic claim that such a logic is not effective since the set *Val* of valid formulas is not effective at all (see for example [7]). At the same time it is possible to prove that the L-subset of theorems of a decidable L-theory is semi-decidable and therefore, that the fuzzy set lt of the logically true sentence is semi-decidible (see [1]). This apparent contrast depends on the fact that *Val* is a cut of lt and, as claimed in Theorem 3, it is not surprising that lt is semi-decidable and that such a cut is not recursively enumerable.

6 Fuzzy machines and fuzzy grammars

A basic question is whether our definition of recursive enumerability is the correct formal counterpart of the intuition and experience of fuzzy people about fuzzy computability. In other words:
Is our definition a reasonable proposal for a "Church Thesis" in multi-valued logic ?

As an attempt to face this question, we consider the notions of *fuzzy Turing machine* and *fuzzy grammar* given in literature. Then, we assume that in the effective lattice L an operation $\otimes$ is defined to interpret the conjunction and that $\otimes$ is order-preserving, associative, commutative and such that $x \otimes 1 = x$ for every $x \in L$. We assume also that $\otimes$ is recursive on the basis B. These conditions are satisfied in all the main multi-valued logics. Firstly, we recall the notion of fuzzy grammar (see [8] and ([9]))

Definition 7. *An L-grammar is a structure $G = (T, I, \mu, s)$ where:*
- T is a finite set and $I \subset T$,
- $\mu : T^+ \times T^+ \to B$ is a finite L-subset (the L-subset of productions)
- $s \in T - I$ (the start symbol).

Given $\lambda \neq 0$ and two words w, w', we say that w' is *directly derivable from w with degree λ* if $x, y \in T^+$ and $a, b \in T^*$ exists such that $w = axb$, $w' = ayb$ and $\lambda = \mu(x, y)$. We say that a sequence $\pi = (w_1, ..., w_p, \lambda_1, ..., \lambda_{p-1})$ is a *derivation for w at degree $\lambda(\pi) = \lambda_1 \otimes ... \otimes \lambda_{p-1}$* provided that $w_1 = s$, $w_p = w$ and, for $i = 1, 2, ..., p-1$, the word w_{i+1} is directly derivable from w_i with degree λ_i. Since it is possible that there are different derivations of the same word, the L-language generated by an L-grammar is defined as follows.

Definition 8. *Let $G = (T, I, \mu, s)$ be an L-grammar, then the L-language generated by G is the L-subset $s : I^+ \to L$ defined by*

$$s(w) = \sup\{\lambda(\pi) : \pi \text{ is a derivation of } w\}. \tag{4}$$

There are various attempts to formalize of the notion of fuzzy algorithms in terms of Turing machines. The first ones are dated in late 1960s when this notion was introduced by L. A. Zadeh (see [16]). The following definition is an obvious extension of the one proposed by E. S. Santos in [13] (see also [15]).

Definition 9. *An L-Turing machine is a structure $F = (S, T, I, b, q_0, q_f, \mu, \otimes)$, where*
- *S is the finite set of states;*
- *T is the finite set of tape symbols;*
- *$I \subset T$ is the set of input symbols ;*
- *μ is an L-subset of $S \times T \times S \times T \times \{-1, 0, 1\}$ with values in B (we call L-transition function)*
- *$b \in T - I$ is the blank symbol;*
- *q_0 and q_f are the initial and accepting states, respectively.*

Symbol -1 (+1) denotes a move by one cell to the left (right) and 0 denotes no move. The tape symbols can be printed on a tape that has a left-most cell but is unbounded to the right. A *move* is an element $m = (s_1, t_1, s_2, t_2, d)$ in $S \times T \times S \times T \times \{-1, 0, 1\}$ and this move is realized provided that if the current state is s_1 and the tape symbol scanned by the machine's head is t_1, then F will enter the new state s_2, the new tape symbol t_2 will rewrite the previous symbol t_1, and the tape head will move in accordance with d. The value $\mu(m)$ is a valuation of correctness (possibility) of the move m. The notion of computation is defined as usual with the help of instantaneous descriptions (IDs). An instantaneous description Q_t of F working on input w at time t is a unique description of the machine's tape, of its state and of the position of the machine's head after performing its tth move on input w.

Definition 10. *If Q_t and Q_{t+1} are two IDs we denote by $D(Q_t, Q_{t+1})$ the last upper bound of the set of correctness degrees $\mu(m)$ of the moves m leading from Q_t to Q_{t+1}.*

We can interpret $D(Q_t, Q_{t+1})$ as the valuation in a multi-valued logic of the claim "there is a correct move leading from Q_t to Q_{t+1}". Observe that if no move exists leading from Q_t to Q_{t+1} then $D(Q_t, Q_{t+1}) = 0$, otherwise $D(Q_t, Q_{t+1})$ is a maximum and we can calculate it in an effective way. On input w whose length is n, the machine starts its computation in an initial ID, we denote by $Q(w)$, describing the tape holding a string of n input symbols (the so-called input string, or input word), one symbol per cell starting with the leftmost cell. All cells to the right of the input string are blank. The head is scanning the leftmost cell and the current state is q_0. From this ID the computation proceeds to an ID, we denote by Q_1 which is reachable in one step from Q_0, etc.

Definition 11. *A computation is a sequence $Q_0, ...Q_k$ of IDs. We extend the function D to any computation $Q_0, ..., Q_k$, by setting*

$$D(Q_0, ..., Q_k) = D(Q_0, ..., Q_{k-1}) \otimes D(Q_{k-1}, Q_k).$$

Moreover, if Q and Q^* are two IDs, we set

$$d(Q, Q^*) = \sup\{D(Q_0, Q_1, ..., Q_t) : Q_0 = Q, Q_t = Q^*\}.$$

We can interpret $D(Q_0, ..., Q_k)$ as the valuation in a multi-valued logic of the claim "the computation $Q_0, ...Q_k$ is correct" and $d(Q, Q^*)$ as the valuation of the claim "there is a correct computation leading from Q to Q^*".

Definition 12. *Let F be an L-Turing machine and $w \in I^+$. Then we say that $Q_0, Q_1...Q_k$ is an accepting computation for w, if $Q_0 = Q(w)$ and Q_k is an ID containing the accepting state q_f. Moreover, the L-language accepted by F is the L-subset $e : I^+ \to L$ of I^+ defined by setting*

$$e(w) = \sup\{d(Q(w), Q^*) : Q^* \text{ is an accepting ID for } w\}. \tag{5}$$

The following theorem shows that the notion of effectiveness for multi-valued logic proposed in this paper is in accordance with just given notions of L-grammar and L-Turing machine.

Theorem 4. *Let s be an L-language either generated by an L-grammar or accepted by an L-Turing machine. Then s is a semi-decidable L-subset.*

Proof. Assume that s is generated by an L-grammar and therefore that s satisfies (4). Then, since for every input w we can enumerate in an effective way the class of derivations for w, s is semi-decidable. A similar argument holds true for the L-Turing machines.

The following theorem shows that, in the case L is a finite chain, the domain-based, the grammar-based and the machine-based notions of effectiveness all coincide.

Theorem 5. *Assume that L is a finite chain and let s be an L-subset. Then the following are equivalent:*
- s is semi-decidable,
- there is a suitable L-grammar able to generate s,
- there is a suitable L-Turing machine able to accept s.

Proof. Let L be the finite chain whose elements are $0 = \lambda_0 < ... < \lambda_n = 1$ and assume that s is semi-decidable. Then all the cuts $C(s, \lambda_i)$ of s are recursively enumerable. For every $0 < i \leq n$, let $G_i = (T, I, M_i, s)$ be a grammar able to generate $C(s, \lambda_i)$ where, as usual, $M_i \subseteq T^+ \times T^+$. Denote by G the L-grammar (T, I, μ, s) obtained by setting $\mu(x) = \sup\{\lambda_i : x \in M_i\}$ and assume that $\otimes$ is the minimum. Then it is easy to see that the L-language generated by such a machine coincides with s.

Likewise, denote by F_i a Turing machine $(S, T, I, b, q_0, q_f, M_i)$ able to accept $C(s, \lambda_i)$ where $M_i \subseteq S \times T \times S \times T \times \{-1, 0, 1\}$. Let F be the L-Turing machine $(S, T, I, b, q_0, q_f, \mu, \wedge)$ such that $\mu(x) = \sup\{\lambda_i : x \in M_i\}$. The it is easy to see that the L-subset accepted by F coincides with s.

In order to examine the case L infinite, it is useful the following proposition whose proof is in [15].

Proposition 9. *Let $(M, \otimes, 1)$ be a finitely generated sub-monoid of $(L, \otimes, 1)$. Then every nonempty subset of M admits a maximal element. If L is totally ordered, every nonempty subset of M admits a maximum. As a consequence, for every word w, the supremum in (4) and in (5) is a maximum.*

The following theorem shows that the L-languages generated by an L-grammar (accepted by an L-Turing machine) satify very particular properties.

Theorem 6. *Assume that L is totally ordered and let s be an L-language generated by an L-grammar (accepted by an L-Turing machine). Then the values assumed by s are in B and all the closed λ-cuts with $\lambda \in B$ are recursively enumerable .*

Proof. Assume that s is generated by an L-grammar and therefore, by Proposition 9, that $s(w) = \max\{\lambda(\pi) : \pi \text{ is a derivation of } w\}$. Then,

$C(s, \lambda) = \{x \in S : \text{there is a derivation } \pi \text{ such that } \lambda(\pi) \geq \lambda\}$

and therefore, to prove that $C(s, \lambda)$ is recursively enumerable, it is sufficient to observe that that the map $\lambda(w)$ is effectively computable and that the relation $\lambda(\pi) \geq \lambda$ is decidable.

In the case of the L-Turing machines we can go on in a similar way.

Such a theorem cannot be extended to the case L is an infinite chain. For example, if $L = U$, then every semi-decidable L-language assuming irrational values gives an example of semi-decidable L-subset such that there is no L-grammar able to generate it and no L-Turing machine able to accept it. A more interesting example is furnished in the following theorem.

Theorem 7. *Assume that L is the effective lattice defined by the interval U and let $big : I^+ \to U$ be the fuzzy subset of the "big words" defined by setting*

$$big(w) = 1 - 1/length(w) \tag{6}$$

for every word w. Then big is a decidable L-language such that
- big assumes only rational values
- the cuts of big are all decidable
- no L-grammar is able to generate big
- no L-Turing machine is able to accept big.

Proof. The proof is trivial. We observe only that, since there is no maximum in the co-domain of *big*, no L-grammar is able to generate *big* and no L-Turing machine is able to accept *big*.

Since should be hard to deny that *big* is decidable from an intuitive point of view, we can conclude that the existing definitions of L-grammar and L-Turing machine are not adequate. This corroborates the domain-based definition of the effectiveness for multi-valued logic and a formulation of the following *"Church Thesis"* for fuzzy set theory:

"the domain-based definitions give the adequate formalization of the intuition and experience of fuzzy people about the effectiveness in the fuzzy framework."

Once we accept such a thesis, it is an open question to find adequate definitions of multi-valued Turing machine and multi-valued grammar.

References

1. Biacino L., Gerla G.: Fuzzy logic, continuity and effectiveness, Archive for Mathematical Logic, **41** (2002) 643–667.
2. Gerla G.: Decidability, partial decidability and sharpness relation for L-subsets, Studia Logica, **46** (1987) 227–238.
3. Gerla G. : Effectiveness and Multivalued Logics, Journal of Symbolic Logic, **71** (2006) 137–162.
4. Gerla G. : Fuzzy logic: Mathematical tools for approximate reasoning, Kluwer Academic Publishers, Dordrecht 2001.
5. Goguen J. A.: The logic of inexact concepts, Synthese, **19** (1968/69) 325–373.
6. Gottwald S.: A treatise on many-valued logics, Research Studies Press, Baldock, 2000.
7. Hájek P.: Metamathematics of fuzzy logic, Kluwer Academic Publishers, Dordrecht, 1998.
8. Lee E. T., Zadeh L. A., Note on fuzzy languages, Information Science, **1** (1969) 421-434.
9. Mizumoto M., Toyoda J., Tanaka K., N-fold fuzzy grammars, Information Science **5** (1973) 25-43.
10. Montagna F.: Three complexity problems in quantified fuzzy logic, Studia Logica, **68** (2001) 143–152.
11. Mundici D., Cignoli R., D'Ottaviano I.: Algebraic foundations of many-valued reasoning, Kluwer Academic Publishers, Dordrecht, 2000.
12. Pavelka J.: On fuzzy logic I: Many-valued rules of inference, Zeitschrift für Mathemathische Logik und Grundlagen der Mathematik, **25** (1979) 45–52.
13. Santos E.S.: Max-Product Machines, J. of Math. Anal. Appl., **37** (1972) 677-686.
14. Smyth M.: Effectively given domains, Theoretical Computer Science, **5** (1977) 257–274.
15. Wiedermann J.: Fuzzy Turing Machines revised, Inform. and Computing, **21** (2002) 1–13.
16. Zadeh L. A.: Fuzzy algorithms, Information and Control, **5** (1968) 94–62.
17. Zadeh L. A.: The concept of a linguistic variable and its application to approximate reasoning I, II, III, Information Sciences, **8**, **9** (1975) 199–275, 301–357, 43–80.