Available online at www.sciencedirect.com
ScienceDirect
Fuzzy Sets and Systems 159 (2008) 1072 – 1083
www.elsevier.com/locate/fss

On the computing power of fuzzy Turing machines

Benjamín Callejas Bedregal[a,*], Santiago Figueira[b]

[a] *Departamento de Informática e Matemática Aplicada, Laboratório de Lógica e Inteligência Computacional, Universidade Federal do Rio Grande do Norte, Campus Universitário s/n, Lagoa Nova, Natal-RN, CEP 59.072-970, Brazil*
[b] *Departamento de Computación, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires, Pabellón I, Ciudad Universitaria (C1428EGA), Buenos Aires, Argentina*

Received 25 April 2007; received in revised form 25 October 2007; accepted 25 October 2007
Available online 11 December 2007

Abstract

We work with fuzzy Turing machines (FTMs) and we study the relationship between this computational model and classical recursion concepts such as computable functions, recursively enumerable (r.e.) sets and universality. FTMs are first regarded as *acceptors*. It has recently been shown by J. Wiedermann that *these machines have more computational power than classical Turing machines*. Still, the context in which this formulation is valid has an unnatural implicit assumption. We settle necessary and sufficient conditions for a language to be r.e., by embedding it in a fuzzy language recognized by a FTM. We do the same thing for n-r.e. set. It is shown that there is no universal fuzzy machine, and "universality" is analyzed for smaller classes of FTMs. We argue for a definition of computable fuzzy function, when FTMs are understood as *transducers*. It is shown that, in this case, our notion of computable fuzzy function coincides with the classical one.

Keywords: Fuzzy Turing machine; Fuzzy function; Fuzzy set; Universal machine; Recursively enumerable set

1. Introduction

Classical computability admits several but equivalent models. Still, the fuzzification of these models may imply different and nonequivalent concepts of fuzzy computability. Even the same model can be fuzzified in several ways. These facts turn this subject very complex and interesting. A precursor of fuzzy computability was the proper founder of fuzzy set theory, Lotfi Zadeh, who in [38] defines the notion of fuzzy algorithm based on a fuzzification of Turing machines and Markov algorithms. However, that work was not deep enough in the recursion theoretical aspects of the mentioned models. Afterward, Lee and Zadeh in [21] follow the same setting and Santos in [31,32] proves that these two fuzzy models are equivalent. Unfortunately the research in this subject was not continued for more than a decade, revisited only in the works of Harkleroad [16] (for other related works, see for example [6,3,25,11,26,4,27,12]). More recently, with the increasing interest in extrapolating Church–Turing thesis considering other aspects (for example interactions [14,7], real values [35], quantum universe [9], etc.), the research on fuzzy computability has gained new strength, mainly because Wiedermann [36,37] claimed that it is possible to solve the halting problem (more precisely, it is possible to accept recursively enumerables r.e., sets and co-r.e. sets) in a class of fuzzy Turing machines (FTMs).

* Corresponding author.
E-mail addresses: bedregal@dimap.ufrn.br (B.C. Bedregal), sfigueir@dc.uba.ar (S. Figueira).

0165-0114/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.fss.2007.10.013

In Section 2 we settle some preliminaries and Section 3 is devoted to introducing nondeterministic Turing machines (NTMs), and to fixing notation to be extended later to the fuzzy context.

In Section 4.1 we work with FTMs, when regarded as *acceptors*. We carefully analyze Wiedermann's statement mentioned above about the computational power of FTMs. We state it in a more rigorous manner and in Theorem 8 we impose necessary and sufficient conditions for a set to be r.e. in terms of associated fuzzy languages recognizable by FTMs (with computable t-norm). We also show that Wiedermann's statement is not completely correct since there are FTMs which could also "recognize" (in the same sense used by Wiedermann) *n*-r.e. [34] sets (and it is well known that for greater and greater values for *n*, these sets may be more complex than the r.e. or co-r.e. ones). In Theorem 9 we characterize the class of *n*-r.e. sets in terms of associated fuzzy languages recognized by FTMs.

In Section 4.2 we deal with the recursive theoretical notion of universality. Theorem 11 shows that there is no universal fuzzy machine for the class of all FTMs. We consider some other narrower classes of fuzzy machines for which we have fuzzy universality.

In Section 4.3, we change the optic and we regard FTMs as *transducers*, i.e. as fuzzy devices computing functions, instead of just recognizing languages. We argue for a definition of fuzzy computable function, when this optic is taken, and in Theorem 12 we show that our proposed notion coincides with the classical one.

2. Elements of fuzzy theory

Let $\mathcal{I}$ be the unitary closed interval, i.e. [0, 1]. A *fuzzy* set A in an universe U_A (a classical set) is a function

$$\mu_A : U_A \to \mathcal{I}.$$

Thus, for each $x \in U_A$, $\mu_A(x)$ provides the *belonging degree* of the element x in the fuzzy set A. For each fuzzy set A, we define its *support* set as

$$S(A) = \{a \in U_A : \mu_A(a) > 0\}$$

and its *crisp* set as

$$C(A) = \{a \in U_A : \mu_A(a) = 1\}.$$

2.1. *t-norms*

Triangular norms, or simply t-*norms*, were originally introduced by Menger in [24] to model the distance in probabilistic metric spaces. But the axiomatic definition of t-norm used today was given by Schweizer and Sklar in [33]. Nevertheless, Alsina et al. [1] showed that this notion could be adequate to model the conjunction in fuzzy logics or equivalently the intersection of fuzzy sets. A t-*norm* on $\mathcal{I}$ is any commutative and associative mapping $T : \mathcal{I} \times \mathcal{I} \to \mathcal{I}$ such that 1 is the neutral element and T is monotonic with respect to the natural order on $\mathcal{I}$. Sometimes t-norms will be used in infix notation instead of the functional form. In this case, we will usually write the symbol $*$. Classical examples of t-norms are the following: $G(x, y) = \min\{x, y\}$ (Gödel t-norm), $P(x, y) = xy$ (product t-norm) and $Ł(x, y) = \max\{x + y - 1, 0\}$ (Łukasiewicz t-norm).

An element $z \in (0, 1)$ is said to be a *zero divisor* of a t-norm $*$ if there exists $y \in (0, 1)$ such that $y * z = 0$. For example, each $z \in (0, 1)$ is a zero divisor of $Ł$.

Since a t-norm is a function on uncountable sets, the notion of computable t-norm is not the classical one. There are several computability theories for uncountable sets (see for example [15,5,29,13,2,35]) and therefore, several suitable ways of defining computable t-norms. In the following, we will consider a definition based on the domain theory point of view [2].

A t-norm T is *computable* if $T : (\mathcal{I} \cap \mathbb{Q}) \times (\mathcal{I} \cap \mathbb{Q}) \to \mathcal{I} \cap \mathbb{Q}$ is a computable function in the usual sense and for each $x, y \in \mathcal{I}$, $T(x, y) = \sup\{T(q, p) : q < x, p \leqslant y \text{ and } q, p \in \mathcal{I} \cap \mathbb{Q}\}$.

Notice that, in this case being computable implies being continuous. Wiedermann in [37] does not require the preservation of suprema, and therefore his computable t-norms are not necessarily continuous. Continuity is necessary to guarantee an approximation process for any possible degree value.

2.2. Fuzzy functions

Following Dubois and Prade in [10], "under the name *fuzzy functions* are gathered various kinds of mappings between sets generalizing ordinary mapping in some sense... A fuzzy function can be understood in several ways according to where fuzziness occurs" and according to which aspect of the crisp function is taken into account by the fuzzy function. Although there are several notions of fuzzy functions in the literature (for example [28,10,20,8,26,39,30,22]), they all can be classify in three kinds [10]:

- Fuzzily constrained functions.
- Fuzzy extension of a nonfuzzy function.
- Fuzzy extension of a nonfuzzy variable.

The partial version of the first one of those notions may be formalized as follows:

Let A and B be fuzzy sets. A classical partial function $f : U_A \to U_B$ is a *fuzzy partial function* from A to B, if

$$\forall x \in U_A, \quad f(x)\uparrow \quad \text{or} \quad \mu_A(x) \leqslant \mu_B(f(x)). \tag{1}$$

Notice that it is possible to find many natural and simple examples of fuzzy partial functions, i.e. cases where the membership degree of $f(x)$ to the fuzzy set B increase with respect to the membership degree of x to the fuzzy set A. Nevertheless, it is also possible to find several natural examples where it occurs just the opposite, i.e. situations where the membership degree of $f(x)$ to the fuzzy set B decrease with respect to the membership degree of x to the fuzzy set A. As can be seen in the next sections, by a property of the t-norm ($T(x, y) \leqslant \min(x, y)$), the function computed by a FTM never increases the membership degree of their inputs. Thus, in this context, a more reasonable notion of fuzzy function would be the dual of the partial fuzzy function notion in Eq. (1).

Let A and B be fuzzy sets. A classical partial function $f : U_A \to U_B$ is a *dual-fuzzy partial function* from A to B, if

$$\forall x \in U_A, \quad f(x)\uparrow \quad \text{or} \quad \mu_B(f(x)) \leqslant \mu_A(x). \tag{2}$$

Clearly, composition of dual-fuzzy partial functions are dual-fuzzy partial functions.

Let f be a dual-fuzzy partial function. We define the partial function $S(f) : S(A) \to S(B)$ as the *support* of f, and the partial function $C(f) : C(A) \to C(B)$ as the *crisp* of f in the following way:

$$S(f)(x) = \begin{cases} f(x) & \text{if } \mu_B(f(x)) > 0, \\ \uparrow & \text{otherwise,} \end{cases} \qquad C(f)(x) = \begin{cases} f(x) & \text{if } \mu_B(f(x)) = 1, \\ \uparrow & \text{otherwise.} \end{cases}$$

3. Nondeterministic Turing machines

In the literature one can find diverse definitions of NTMs and all of them are equivalent (see for example [17,19,23]). We use the following definition: A NTM is a septuple $\mathcal{T} = \langle Q, \Sigma, \Gamma, \delta, q_0, \Box, F\rangle$ where Q is a set of states, Σ is the input alphabet, Γ is the tape alphabet, $q_0 \in Q$ is the starting state, $\Box \in \Gamma$ is the blank symbol, $F \subseteq Q$ is the set of final states and $\delta \subseteq Q \times \Gamma \times Q \times \Gamma \times \{R, L\}$ is the set of instructions, i.e. the "next move" relation.

We will use the following string functions: $head(w)$ returns the leftmost symbol of $w\Box$ (in case w is the empty word, the result is $\Box$); $head^R(w)$ returns the rightmost symbol of $\Box w$; $tail(w)$ returns the string w without its leftmost symbol, if w is not empty and the empty word otherwise; $tail^R(w)$ returns the string w without its rightmost symbol, if w is not empty and the empty word otherwise.

An *instantaneous description* (ID) of a NTM, ID for short, is a triple (u, q, v) meaning that the tape content is the string $\Box^{\omega^*}uv\Box^{\omega}$, the current state is q and the head is pointing at the leftmost symbol of $v\Box^{\omega}$. For notational simplicity we will omit the parentheses and comma of IDs. A *valid move* from an ID uqv into an ID $u'pv'$ in the NTM $\mathcal{T}$, denoted by $uqv \vdash_{\mathcal{T}} u'pv'$, occurs whenever

$$\exists (q, head(v), p, b, R) \in \delta \text{ such that } u' = u \circ b \text{ and } v' = tail(v) \quad \text{or}$$

$$\exists (q, head(v), p, b, L) \in \delta \text{ such that } u' = tail^R(u) \text{ and } v' = head^R(u) \circ b \circ tail(v).$$

As usual, an ID $u'pv'$ is reached from an ID uqv, denoted by $uqv \vdash^*_{\mathcal{T}} u'pv'$, if $uqv = u'pv'$ or there exists an ID $u''rv''$ such that

$$uqv \vdash_{\mathcal{T}} u''rv'' \quad \text{and} \quad u''rv'' \vdash^*_{\mathcal{T}} u'pv'.$$

When a NTM $\mathcal{T}$ is regarded as an *acceptor*, we say that the string $w \in \Sigma^*$ is accepted by $\mathcal{T}$ if $q_0 w \vdash^*_{\mathcal{T}} uq_f v$ for some $u, v \in \Gamma^*$ and $q_f \in F$. As usual the *language accepted* by a NTM $\mathcal{T}$, denoted by $L(\mathcal{T})$, is the set of all strings accepted by $\mathcal{T}$.

4. Fuzzy Turing machines

Zadeh [38], Lee [21] and Santos [31] introduced the model of FTMs and the languages accepted by this kind of machines, i.e. a class of fuzzy languages. Classical languages are linked to fuzzy languages through the support and crisp part of a fuzzy set. It turns out that this fuzzy machine model is computationally too powerful: in [37], Wiedermann claims that, in fact, its nondeterministic version accepts non-r.e. languages and that they can solve undecidable problems (these assertions will be fully analyzed in Section 4.1). On the other hand, the model is too restrictive from a fuzzy logic point of view, since it only considers the Gödel t-norm. The idea of this FTM is to establish an *uncertainty degree* for the acceptance of a given string or, analogously, the *membership degree* of the string to the language. In order to compute this degree from individual degrees, a composition on the t-norm evaluation is used. Wiedermann [36,37] introduced the class of FTMs as a fuzzy extension of the NTMs, where each transition has a membership degree associated to it. In this case, he worked with arbitrary t-norms for the evaluation. We consider this same kind of FTMs:

Definition 1. A FTM is a triple $\mathcal{F} = \langle \mathcal{T}, *, \mu \rangle$ where $\mathcal{T} = \langle Q, \Sigma, \Gamma, \delta, q_0, \Box, F \rangle$ is a NTM, $*$ is a t-norm and μ is a map which assigns a membership degree to each tuple in the "next move" relation δ, i.e. $\mu : \delta \to \mathcal{I}$.

An ID of a FTM $\mathcal{F} = \langle \mathcal{T}, *, \mu \rangle$ is a pair (uqv, d) where uqv is an ID of the NTM $\mathcal{T}$, i.e. uv is the string in the tape, the head is pointing to the leftmost symbol of v, the current state is q and d is the membership degree accumulated up to this moment.

A *valid move* from an ID (uqv, d) into an ID $(u'pv', d')$, denoted by $(uqv, d) \vdash_{\mathcal{F}} (u'pv', d')$, occurs whenever $uqv \vdash_{\mathcal{T}} u'pv'$ and

$$d' = \begin{cases} d * \mu(q, head(v), p, head^R(u'), R) & \text{if } tail^R(u') = u, \\ d * \mu(q, head(v), p, head(tail(u')), L) & \text{if } tail^R(u) = u'. \end{cases}$$

As with the NTM case, an ID $(u'pv', d')$ is reached from an ID (uqv, d), denoted by $(uqv, d) \vdash^*_{\mathcal{F}} (u'pv', d')$, if $(uqv, d) = (u'pv', d')$ or there exists an ID $(u''rv'', d'')$ such that $(uqv, d) \vdash_{\mathcal{F}} (u''rv'', d'')$ and $(u''rv'', d'') \vdash^*_{\mathcal{F}} (u'pv', d')$.

4.1. FTMs as acceptors

The *degree of acceptance* in a FTM $\mathcal{F}$ of a string w is

$$\deg_{\mathcal{F}}(w, k) = \sup\{d \in \mathcal{I} : (q_0 w, k) \vdash^*_{\mathcal{F}} (uq_f v, d) \text{ for some } q_f \in F\}.$$

When $k = 1$ we will omit it and we will write $\deg_{\mathcal{F}}(w)$.

Since a language is just a set of strings, a natural definition for fuzzy language is "a fuzzy set of strings". Thus, the fuzzy language *accepted* by a FTM $\mathcal{F}$ is

$$L(\mathcal{F}) = \{(w, \deg_{\mathcal{F}}(w)) : w \in \Sigma^*\}.$$

Thus, $L(\mathcal{F})$ is a fuzzy set with universe Σ^* and membership function being the function $\deg_{\mathcal{F}}$.

In this article, we work with FTMs with rational degree membership and computable t-norm.

Definition 2. Let C be the class of all FTMs with rational degree membership and computable t-norm, i.e. fuzzy machines $\mathcal{F} = \langle \mathcal{T}, *, \mu \rangle$ where the range of μ is $\mathbb{Q} \cap \mathcal{I}$ and the t-norm $*$ is computable.

For any set $A \subseteq \Sigma^*$, we define $A(w) = 1$ if $w \in A$ and $A(w) = 0$ otherwise. Recall that a set $A \subseteq \Sigma^*$ is r.e. when there is a computable approximation $g: \Sigma^* \times \mathbb{N} \to \{0, 1\}$ such that

$$\lim_{t\to\infty} g(w, t) = A(w), \tag{3}$$

$$(\forall w \in \Sigma^*)\ g(w, 0) = 0, \tag{4}$$

$$\#\{t : g(w, t) \neq g(w, t + 1)\} \leqslant 1. \tag{5}$$

Condition (3) says that the approximation g eventually stabilizes in 1 or 0 depending whether $w \in A$ or $w \notin A$, respectively. Condition (4) says that the approximation starts in 0 and condition (5) says that for any $w \in \Sigma^*$, $g(w, \cdot)$ can change at most once. A set is *co-r.e.* if its complement is r.e.

In [36,37], Wiedermann claims that FTMs can solve undecidable problems and that the languages accepted by these machines (when we consider a computable t-norm) are exactly the union of r.e. sets and co-r.e. sets. Evidently there is some abuse in this terminology, since r.e. sets are *ordinary* languages and the languages accepted by FTMs are *fuzzy* languages. Hence, there is some kind of implicit *fuzzification* when he says that FTMs accept noncomputable r.e. sets. This fuzzification is a way of transforming or embedding an ordinary set into a fuzzy set, exploiting the degree of acceptance to somehow codify the membership of each element of the set.

To explain what is the exact assertion of Wiedermann, let us first define a special way of fuzzifying ordinary sets into fuzzy sets. For any language A and for rationals a and b ($a, b \in \mathcal{I}$) we define the following fuzzification of the set A:

$$F_A(a, b) = \{(w, a): w \in A\} \cup \{(w, b): w \notin A\}.$$

The following theorem is the essence of what Wiedermann proves in [37, Theorem 3.1]:

Theorem 3. *Let $A \subseteq \Sigma^*$ and $0 \leqslant b < 1$:*

(1) *if A is r.e. then $F_A(a, 1)$ is a language of some* FTM;
(2) *if A is co-r.e. then $F_A(1, a)$ is a language of some* FTM.

Even more, we can prove the following stronger result:

Theorem 4. *Let $A \subseteq \Sigma^*$ be any set and let $a, b \in \mathbb{Q}$ such that $0 \leqslant b < a \leqslant 1$. The following are equivalent*:

(1) *A is* r.e.;
(2) *there is some* FTM *in* C *which accepts the fuzzy language $F_A(a, b)$.*

Proof. ($1 \Rightarrow 2$) Let $g: \Sigma^* \times \mathbb{N} \to \{0, 1\}$ be the computable approximation of A. Let $\mathcal{F}$ be the FTM which on input w, it has a nondeterministic branch starting from state q_0:

- $\mathcal{F}$ passes from q_0 to the final state q_f via a transition with degree b, and
- $\mathcal{F}$ passes from q_0 to a procedure which scans $g(w, 0), g(w, 1), \ldots$ until it finds some t such that $g(w, t) = 1$ (all this procedure is carried on with transitions of degree 1). If this ever happens then $\mathcal{F}$ goes to the final state q_f via a transition with degree a and otherwise it keeps on searching (so it never reaches the final state).

Now, if $w \in A$ then there is a least s such that $g(w, s) = 1$, so there will be two accepting paths in $\mathcal{F}$: the one coming from the first nondeterministic branch, with accepting degree b, and the one coming from the second nondeterministic branch, with accepting degree a. Since $a > b$ then $(w, a) \in L(\mathcal{F})$. On the other hand, if $w \notin A$ then there is only one accepting path in the execution of $\mathcal{F}$—the one coming from the first nondeterministic branch—and hence $(w, b) \in L(\mathcal{F})$.

($2 \Rightarrow 1$) Suppose $\mathcal{F} = \langle \mathcal{T}, *, \mu \rangle \in \mathsf{C}$ is a FTM which accepts $F_A(a, b)$. We define $g: \Sigma^* \times \mathbb{N} \to \{0, 1\}$ in the following way $g(w, t) = 1$ if by stage t we find that $\mathcal{F}(w)$ arrives to a final state with accepting degree a. Otherwise $g(w, t) = 0$. Observe that g is computable since the t-norm $*$ is computable, and the range of μ is a subset of the rational numbers of $\mathcal{I}$. $\square$

In the above proof, the fuzzification used to interpret an ordinary language into a fuzzy language consists in defining w in the accepted language of $\mathcal{F}$ with membership degree a, for every $w \in A$; and w with a *smaller* membership degree b, for every $w \notin A$. It is worth noting that this result only applies when this particular way of fuzzifying r.e. sets of strings is used—i.e., when working with $F_A(a,b)$. Although one could intuitively think that if there is a FTM which accepts $F_A(a,b)$, then there should be another FTM which accepts a "simple" transformation of $F_A(a,b)$, such as $F_A(b,a)$, the (contra positive version of) following proposition shows that this is not the case.

Proposition 5. *Let $A \subseteq \Sigma^*$ be an r.e. set and let $a, b \in \mathbb{Q}$ such that $0 \leqslant b < a \leqslant 1$. If $F_A(b,a)$ is accepted by some* FTM *of* C *then A is computable*.

Proof. We show that there is an effective decision procedure for testing the membership of any string w to the set A. Here is the procedure: run in parallel the enumeration of A (which exists by hypothesis) and simulate all the execution paths of $\mathcal{F}(w)$. Eventually, either the approximation tells us that $w \in A$, or we find an accepting path of $\mathcal{F}(w)$ with membership degree a. Since $a > b$, then the path that we have found has maximum degree, and hence $w \notin A$. □

The above proposition shows that the fuzzification used by Wiedermann is intrinsically linked to the fact that A is r.e.; the result is not independent of the fuzzification used. Indeed, when Wiedermann [37] considers co-r.e. sets A, he changes the fuzzification, and in this case, he shows that there is a FTM which accepts $F_A(b,1)$, for any fixed rational $b \in [0,1)$. Hence, one has to be careful with Wiedermann's claim "languages *accepted* by FTM with computable t-norm coincide with the class of r.e. sets union co-r.e. sets": the notion of *acceptance* here involves a particular fuzzification, which differs in the r.e. case and the co-r.e. case. In fact, in the proof of [37, Theorem 3.2] there is another point that needs a better justification, since all it shows is that there exist r.e. languages L_1 and L_2 such that $L_1 \backslash L_2 = \{w\#d : (w,d) \in L(\mathcal{F})\}$ for any FTM $\mathcal{F}$. Of course, $L_1 \backslash L_2$ is *difference r.e.*, or equivalently, 2-*r.e.* (see the definition given below Eq. (6)). But as it is well known, in general this does not imply that $L_1 \backslash L_2$ is r.e. or co-r.e. (see for example [34, Exercise 3.7, p. 58]), as it is affirmed in Wiedermann's proof.

We obtain the following corollaries from Theorem 4 and Proposition 5. Both follow immediately from the observation that $F_{\Sigma^* \backslash A}(b,a) = F_A(a,b)$.

Corollary 6. *Let $A \subseteq \Sigma^*$ be a set and let $a, b \in \mathbb{Q}$ such that $0 \leqslant b < a \leqslant 1$. A is co-r.e. iff there is a* FTM *in* C *which accepts the fuzzy language $F_A(b,a)$*.

Thus, A is computable if and only if there are FTMs accepting the languages $F_A(a,b)$ and $F_A(b,a)$, respectively.

Corollary 7. *Let $A \subseteq \Sigma^*$ be co-r.e. set and let $a, b \in \mathbb{Q}$ such that $0 \leqslant b < a \leqslant 1$. If $F_A(a,b)$ is accepted by some* FTM *of* C *then A is computable*.

In fact, it is not necessary to fix the values of the rationals a and b in the above results. Indeed, using the same strategy used in Theorem 4, it is not difficult to prove:

Theorem 8. *The following are equivalent*:

(1) *A is r.e.*;
(2) *for any $a \in \mathbb{Q} \cap (0,1)$ there is some* FTM *$\mathcal{F} \in$* C *such that $w \in A$ iff $\deg_{\mathcal{F}}(w) \in [a,1]$.*

Proof. ($1 \Rightarrow 2$) Follows directly from Theorem 4.

($2 \Rightarrow 1$) Observe that we can simulate all the execution paths of $\mathcal{F}(w)$ in parallel. Whenever we see that $\mathcal{F}$ reaches a final state via an execution path with acceptance degree $d > a$, then $\deg_{\mathcal{F}}(w) \geqslant d > a$ and hence it is safe to assert $w \in A$. This procedure informally describes an effective computable approximation of A. □

So far we have been working with special fuzzifications of r.e. sets (and symmetrically, with co-r.e. sets). What about other sets which can be more complex in terms of computability theory? For any $n \geqslant 1$, one can define the *n-r.e.* [34, p. 58] sets as those for which there is a computable approximation $g : \Sigma^* \times \mathbb{N} \to \{0,1\}$ of A such that conditions (3) and (4) hold and condition (5) is replaced by

$$\#\{t : g(w, t) \neq g(w, t + 1)\} \leqslant n. \tag{6}$$

This just means that for every $w \in \Sigma^*$, $g(w, \cdot)$ makes at most n changes. According to this new definition r.e. sets as described before are just 1-r.e.

A set A is co-n-r.e. if the complement of A is n-r.e. Equivalently, A is co-n-r.e. if there is a computable approximation g of A such that conditions (3) and (6) are true and condition (4) is replaced by

$$(\forall x \in \Sigma^*)\ g(x, 0) = 1.$$

See [34] for more details.

The following is a generalization of Theorem 8 and characterizes (classical) n-r.e. sets in terms of the existence of specific fuzzy sets accepted by FTMs.

Theorem 9. *The following are equivalent*:

(1) *A is n-r.e.*;
(2) *For any $a_0, a_1, \ldots, a_{n-1} \in \mathbb{Q} \cap \mathcal{I}$ such that $0 < a_0 < a_1 < \cdots < a_{n-1} < a_n = 1$, there is some* FTM $\mathcal{F} \in \mathsf{C}$ *such that $w \in A$ iff $\deg_{\mathcal{F}}(w) \in \bigcup_{0 \leqslant i < n/2} [a_{2i}, a_{2i+1}]$.*

Proof. ($1 \Rightarrow 2$) Suppose $g\colon \Sigma^* \times \mathbb{N} \to \{0, 1\}$ is a computable approximation of A such that (3), (4) and (6) are true. Also, assume $a_0, a_1, \ldots, a_n$ as in item 2 of our statement. Define $a_{-1} = 0$ and let $b_i = (a_{2i} + a_{2i+1})/2$ for $i \in \{0, \ldots, \lceil n/2 \rceil - 1\}$ and $c_j = (a_{2j-1} + a_{2j})/2$ for $j \in \{0, \ldots, \lfloor n/2 \rfloor\}$. Consider the following FTM $\mathcal{F}$ whose behavior on input $w \in \Sigma^*$ is described by the following procedures:

- **Initial**: nondeterministically go to procedure P_i, with membership degree b_i for $i \in \{0, \ldots, \lceil n/2 \rceil - 1\}$ and to procedure N_j with membership degree c_j for $j \in \{0, \ldots, \lfloor n/2 \rfloor\}$.
- **Procedure** P_i: search for an $p_i \geqslant 0$ such that

$$\#\{t : t \leqslant p_i, g(w, t) \neq g(w, t + 1)\} = 2i + 1.$$

  Then go to the final state with membership degree 1. If such p_i is never found, then it gets undefined.
- **Procedure** N_j: If $j = 0$, go to the final state. Else search for an $n_j \geqslant 0$ such that

$$\#\{t : t \leqslant n_j, g(w, t) \neq g(w, t + 1)\} = 2j.$$

  Then go to the final state with membership degree 1. If such n_j is never found, then it gets undefined.

Clearly $\mathcal{F}$ is in C. The idea behind this machine is that procedure P_i bets that $w \in A$ and that the approximation $g(w, \cdot)$ makes $2i + 1$ changes. Reciprocally, procedure N_j bets that $w \notin A$ and $g(w, \cdot)$ makes $2j$ changes. Of course neither P_i nor N_j may be sure whether their guess is correct or not. But in case it is incorrect, there will be another procedure (with greater accepting degree) which will be correct. Notice that the accepting degree of all the procedures (if they ever reach a final state) are ordered linearly in the following way:

$$N_0, P_0, N_1, P_1, N_2, P_2, \ldots .$$

This is so because

$$c_0 < b_0 < c_1 < b_1 < c_2 < b_2 \cdots .$$

Now, suppose $x \in A$ and suppose g makes exactly $2k + 1$ changes, i.e.

$$\#\{t : g(x, t) \neq g(x, t + 1)\} = 2k + 1.$$

Then, according to $\mathcal{F}$, there will be exactly $k + 1$ accepting paths coming from $P_0, \ldots, P_k$ and $k + 1$ accepting paths coming from $N_0, \ldots, N_k$. Observe that all the other procedures never reach a final state. Notice that procedure P_k induces an accepting path with membership degree $b_k \in [a_{2k}, a_{2k+1}]$ and this is the maximum among $P_0, \ldots, P_k$.

Procedures $N_0, \ldots, N_k$ induce accepting paths of degree less than b_k and therefore $\deg_{\mathcal{F}}(x) = b_k$, which clearly belongs to $\bigcup_{0 \leqslant i < n/2} [a_{2i}, a_{2i+1}]$.

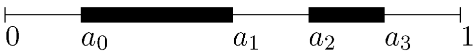
Suppose $x \notin A$ and suppose g makes exactly $2k$ changes, i.e.

$$\#\{t : g(x,t) \neq g(x,t+1)\} = 2k.$$

There will be exactly $k+1$ accepting paths coming from $N_0, \ldots, N_k$ and k accepting paths coming from $P_0, \ldots P_{k-1}$. The greatest degree of acceptance will be the one induced by N_k, which is $c_k \in (a_{2k-1}, a_{2k})$. Therefore $\deg_{\mathcal{F}}(w) = c_k$, which clearly does not belong to $\bigcup_{0 \leqslant i < n/2} [a_{2i}, a_{2i+1}]$.

($2 \Rightarrow 1$) Suppose a FTM $\mathcal{F}$ as in item 2. We can simulate the machine step by step and define g as follows: $g(x, 0) = 0$ for all $x \in \Sigma^*$; $g(x, t+1) = 1$ if the maximum degree of acceptance seen by stage t belongs to $\bigcup_{0 \leqslant i < n/2} [a_{2i}, a_{2i+1}]$ and $g(x, t+1) = 0$ otherwise. When increasing t, the accepting degree cannot decrease and therefore $g(w, \cdot)$ may change at most n times, showing that A is n-r.e. □

The above theorem gives some embedding from any n-r.e. set A into a special fuzzy set. We analyze the accepting degree of this induced fuzzy set. Some regions of accepting degrees will account for *elements* of A and other regions will account for *nonelements* of A. For example, if we take a 3-r.e. set A then we can think of a_0, a_1, a_2 and a fuzzy set accepted by some FTM $\mathcal{F}$ of C such that $w \in A$ if and only if $\mathcal{F}$ accepts w with a membership falling in any of the shaded intervals:



The following is a straight corollary from the previous theorem:

Corollary 10. *The following are equivalent*:

(1) *A is co-n-r.e.*;
(2) *For any $a_0, a_1, \ldots, a_{n-1} \in \mathbb{Q} \cap \mathcal{I}$ such that $a_{-1} = 0 < a_0 < a_1 < \cdots < a_{n-1} < a_n = 1$, there is some* FTM *$\mathcal{F} \in \mathsf{C}$ such that $w \in A$ iff* $\deg_{\mathcal{F}}(w) \in \bigcup_{0 \leqslant i \leqslant n/2} (a_{2i-1}, a_{2i})$.

Observe that there is nothing special in the set

$$\bigcup_{0 \leqslant i < n/2} [a_{2i}, a_{2i+1}] \tag{7}$$

of Theorem 9 for using closed intervals. The same proof can also be carried on with intervals of the form $(a_{2i}, a_{2i+1}]$ in (7). Also, intervals of the form (a_{2i}, a_{2i+1}) or $[a_{2i}, a_{2i+1})$ can be used, provided one adds to the set the real 1 in the case that n is odd. In other words, (7) may also be replaced with

$$\bigcup_{0 \leqslant i < n/2} \langle a_{2i}, a_{2i+1} \rangle \cup C,$$

where $C = \emptyset$ if n is even, and $C = \{1\}$ if n is odd, and where symbol $\langle b, c \rangle$ represents any of the following intervals: (b, c), $(b, c]$, $[b, c)$ or $[b, c]$.

4.2. Universal FTMs

In classical recursion theory, we have the notion of *universal machine*: in short a machine capable to simulate the behavior of every other machine. If $(\mathcal{M}_i)_{i \in \mathbb{N}}$ is an enumeration of all deterministic Turing machines (DTMs) (when seen as transducers), then $\mathcal{U}$ is said *universal* when $\mathcal{M}_i(w)\downarrow$ iff $\mathcal{U}(\langle w, i \rangle)\downarrow$ and if $\mathcal{M}_i(w)\downarrow$ then $\mathcal{M}_i(w) = \mathcal{U}(\langle w, i \rangle)$ (here $\langle \cdot, \cdot \rangle : \mathbb{N} \times \Sigma^* \to \Sigma^*$ is the usual pairing function). We also have a universal machine, when thinking of acceptors. In this case, $(\mathcal{M}_i)_{i \in \mathbb{N}}$ would correspond to an enumeration of all r.e. sets (identifying the domain of $\mathcal{M}_i$ with the ith r.e. set) and $\mathcal{U}$ is said universal when $\mathcal{M}_i(w)\downarrow$ iff $\mathcal{U}(\langle w, i \rangle)\downarrow$.

Since all the elements of each FTM $\in \mathsf{C}$ are finitely representable, we can assign Gödel numbers to each FTM, and obtain $(\mathcal{F}_i)_{i \in \mathbb{N}}$, an enumeration of C.

Following the notion of universality for classical computability, a *fuzzy universal machine* (regarded as an acceptor) $\mathcal{U}_{\mathrm{F}}$ for the class C would be a special fuzzy machine with the ability to simulate the behavior of any other fuzzy machine in C, i.e. $\mathcal{U}_{\mathrm{F}}(\langle i, w\rangle) = \mathcal{F}_i(w)$. This means that for each $i \in \mathbb{N}$ and $w \in \Sigma^*$:

(1) $\mathcal{F}_i(w)\downarrow$ iff $\mathcal{U}_{\mathrm{F}}(\langle i, w\rangle)\downarrow$, and
(2) if $\mathcal{F}_i(w)\downarrow$ then $\deg_{\mathcal{F}_i}(w) = \deg_{\mathcal{U}_{\mathrm{F}}}(\langle i, w\rangle)$.

Although one could think that, as in the classical scenario, there should be such $\mathcal{U}_{\mathrm{F}}$, the following result refutes the idea:

Theorem 11. *There is no universal fuzzy machine for the class* C.

Proof. Suppose $\mathcal{U}_{\mathrm{F}} = \langle \mathcal{U}, *, \mu\rangle$ where $\mathcal{U} = \langle Q, \Sigma, \Gamma, \delta, q_0, \square, F\rangle$ is a FTM as described above. Obviously, any computational path $t_1, \ldots, t_n$ of $\mathcal{U}_{\mathrm{F}}$ ($t_i \in \delta$) will have degree $\mu(t_1) * \cdots * \mu(t_n) \leqslant 1$. Let

$$d = \max(\{\mu(t) : w \in \Sigma^* \wedge t \in \delta \wedge \mu(t) < 1\} \cup \{0\}).$$

Any accepting path containing some $t \in \delta$ with $\mu(t) \leqslant d$ will have degree $\leqslant d$, hence $\mathcal{U}_{\mathrm{F}}$ has no computational path with degree $\tilde{d} \in \mathbb{Q} \cap \mathcal{I}$ such that $d < \tilde{d} < 1$. Now, let $\mathcal{F}$ be a FTM with Gödel number e such that $L(\mathcal{F}) = \{(w, \tilde{d}) : w \in \Sigma^*\}$. Clearly, $\mathcal{U}_{\mathrm{F}}(\langle w, e\rangle) = \mathcal{F}(w)$, so $\mathcal{U}_{\mathrm{F}}$ must accept $\langle w, e\rangle$ with membership degree $\tilde{d}$, and this is impossible. $\square$

However, when we restrict ourselves to a smaller class, we still may have universality. Let D be a class of FTMs. We say that $\mathcal{U}$ is a universal FTM for the class D, when $\mathcal{U}$ is able to simulate any other machine in D, and $\mathcal{U} \in D$.

For example, let $B \subseteq \mathbb{Q} \cap \mathcal{I}$ and let D_B be the class of FTMs $\mathcal{F} = \langle \mathcal{T}, *, \mu\rangle$ such that B is closed under $*$, i.e. $\forall x, y \in B, x * y \in B$. It is not difficult to see that if B is finite, there is a universal fuzzy machine for the class D_B. Informally, if $B = \{b_1, \ldots, b_k\}$, this universal machine would have k special transitions $t_1, \ldots, t_k$ with $\mu(t_i) = b_i$, and will use them to actually pursue the degree of the simulated machine and input.

It is also interesting to observe that a class of FTMs such as D_B, with finite B, is not the only situation where universality is admitted. For example, consider the product t-norm $P(x, y) = xy$ and $B' \subseteq \{2^{-i} : i \in \mathbb{N}^+\}$. We can see that there is a universal machine for the class $D_{B'}$: a universal machine could have a unique special transition t with $\mu(t) = \frac{1}{2}$ to actually obtain any number of B' by successive applications of the t-norm P.

4.3. FTMs as transducers

We know that Turing machines have two roles: as a language acceptor machine and as a function computer (transducer). Hence, we can think of a FTM as a function computer, but with an additional membership degree. That is, it computes a dual-fuzzy partial function from Σ^* into Γ^*, where the input as well as the output have a membership degree. Since it is not clear what role the nondeterminism plays in computing functions [23], in this section we consider only deterministic FTM, denoted DFTM for short. Without loss of generality, we can assume that a DTM for short, has just a unique final state under which the machine halts when reached.

Let $\mathcal{F} = \langle \mathcal{T}, *, \mu\rangle$ be a DFTM. A dual-fuzzy partial function $f : \Sigma^* \to \Gamma^*$ from the fuzzy set A into the fuzzy set B (i.e. Σ^* and Γ^* are the universes of A and B, respectively) is *computed* by $\mathcal{F}$ if f (when seen as a classical partial function) is computed by the DTM $\mathcal{T}$ and if for each w, whenever $f(w)\downarrow$ then

$$\mu_B(f(w)) = \mu_A(w) * \mu(t_1) * \cdots * \mu(t_n), \tag{8}$$

where $t_1, \ldots, t_n$ is the computational path for $q_0 w \vdash^*_{\mathcal{T}} u q_{\mathrm{f}} v$ with $uv = f(w)$ and q_{f} is the final state of $\mathcal{T}$. Since all t-norms always return a value not greater than the original, then the membership degree of $f(w)$ is never larger than the membership degree of w. Clearly, a DFTM computes a dual-fuzzy partial function for each fuzzification of Σ^*.

We say that a DFTM $\mathcal{F}$ S_*-*computes* a partial function $f : \Sigma^* \to \Gamma^*$ if there exists a dual-fuzzy partial function $\widetilde{f}$ computed by $\mathcal{F}$ such that $S(\widetilde{f}) = f$. Analogously, we say that a DFTM $\mathcal{F}$ C-*computes* a partial function $f : \Sigma^* \to \Gamma^*$ if there exists a dual-fuzzy partial function $\widetilde{f}$ computed by $\mathcal{F}$ such that $C(\widetilde{f}) = f$.

Notice that the function S_*-computed by a DFTM $\mathcal{F}$ could change in case a t-norm with zero divisor is used, whereas the function C-computed by $\mathcal{F}$ is the same independently of the t-norm chosen.

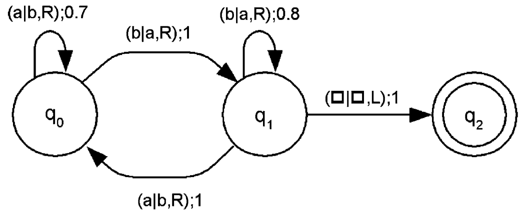


Fig. 1. Example of fuzzy Turing machine.

For example, consider the DFTM with the membership degree at each transition illustrated in Fig. 1. If the Gödel t-norm is used, for the input *aaabbaab* with membership degree 0.7 the machine gives as output *bbbaabba* with membership degree 0.7. Instead, if the Łukasiewicz t-norm is considered, the membership degree of the output is 0.

Theorem 12. *Let $*$ be a t-norm without zero divisors and let $f : \Sigma^* \to \Gamma^*$ be a partial function. The following conditions are equivalent*:

(1) *f is S_*-computable*;
(2) *f is C-computable*;
(3) *f is computable in the classical sense*.

Proof. ($1 \Rightarrow 2$) Let $\mathcal{F} = \langle \mathcal{T}, *, \mu \rangle$ be a DFTM which S_*-computes f. Then, the DFTM $\mathcal{F}' = \langle \mathcal{T}, *, \mu' \rangle$ where for each $t \in \delta$,

$$\mu'(t) = \begin{cases} 1 & \text{if } \mu(t) > 0, \\ 0 & \text{otherwise.} \end{cases}$$

$\mathcal{F}'$ C-computes f, thanks to the nonexistence of zero divisors of $*$.

($2 \Rightarrow 3$) Let $\mathcal{F} = \langle \mathcal{T}, *, \mu \rangle$ be a DFTM which C-computes f, and let

$$\mathcal{T}' = \langle Q, \Sigma, \Gamma, \delta', q_0, \square, F \rangle$$

be the DTM obtained from $\mathcal{T}$ changing the transition relation by: $t \in \delta'$ iff $t \in \delta$ and $\mu(t) = 1$. Clearly, the function computed by $\mathcal{T}'$ is f.

($3 \Rightarrow 1$) Let $\mathcal{T}$ be a DTM which computes f. Then, the DFTM $\mathcal{F} = \langle \mathcal{T}, *, \mu \rangle$, where

$$\mu(t) = \begin{cases} 1 & \text{if } t \in \delta, \\ 0 & \text{otherwise.} \end{cases}$$

S_*-computes (and also C-computes) f. $\square$

Thus, in terms of classical computability S_*-computability and C-computability are equivalent for t-norms without zero divisors. Clearly, the same is valid for languages.

5. Final remarks

The main goal of this paper is not to criticize Wiedermann's work, but rather to clarify the context in which his result is valid. In this sense, we prove that considering the same kind of fuzzification the principal result of Wiedermann [37, Theorems 3.1 and 3.2] is not valid. Other contributions are:

- To provide some results of characterization of n-r.e. sets via FTMs. We first analyzed the case of r.e. sets and later extended this result to n-r.e. sets, which may be more complex in terms of computability theory. We proved that FTMs can also embed this kind of sets in a fuzzy language (in the same way that Wiedermann embedded r.e. sets).
- To prove that it is not possible to achieve a universal FTM. The difficulty comes when we try to simulate the degree of acceptance. It is important to notice that we are not trying to calculate the accepting degree as a written output.

Instead, a universal fuzzy machine should genuinely copy the accepting degree of the simulated FTM, by using its own transitions.

- To give some considerations on the notion of computability of functions by DFTMs and to prove that DFTMs have the same computational power as classical Turing machines (considering two ways of relating these concepts).

As it is well know, the set of valid formulas of a multi-valued logic is not r.e. in the classical sense (see for example [18]). Nevertheless that set is r.e. if we take into account the notion introduced by Gerla in [11] (a proof can be found in [12]). Since FTMs determine a recursively enumerability notion for fuzzy sets which is different from the one given by Gerla, it is reasonable to ask whether the set of valid formulas of a multivalued logic is r.e. or not on that recursively enumerability notion. We will tackle this problem in future work as well as others dealing with the relationship between our results and the ones of Gerla in [11], who provides fuzzifications of several concepts of recursion theory—though some fuzzy notions do not coincide exactly with ours.

Acknowledgments

We thank the anonymous referees for their comments and suggestions which helped to improve the paper.

B.C. Bedregal acknowledges CNPq (Brazilian Research Council) under process numbers 470871/2004-0 and 307879/2006-2. S. Figueira is partially supported by a grant of Fundación YPF, Argentina. He also acknowledges CONICET.

References

[1] C. Alsina, E. Trillas, L. Valverde, On non-distributive logical connectives for fuzzy set theory, Busefal 3 (1980) 18–29.
[2] B.C. Bedregal, B.M. Acióly, A computability on the interval space: a domain approach, in: Proc. XXIII Conf. Latinoamericana de Informática, Vol. I, Valparaíso-Chile, 1997, pp. 107–116.
[3] L. Biacino, G. Gerla, Fuzzy subsets: a constructive approach, Fuzzy Sets and Systems 45 (1992) 161–168.
[4] L. Biacino, G. Gerla, Fuzzy logic, continuity and effectiveness, Arch. Math. Logic 41 (2002) 643–667.
[5] L. Blum, M. Shub, S. Smale, On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines, Bull. Amer. Math. Soc. 21 (1) (1989) 1–46.
[6] B. Clares, M. Delgado, Introduction of the concept of recursiveness of fuzzy functions, Fuzzy Sets and Systems 21 (1987) 301–310.
[7] A.C.R. Costa, G.P. Dimuro, Interactive computation: stepping stone in the pathway from classical to developmental computation, Electronic Notes in Theoretical Computer Science, Vol. 141(5), 2005, pp. 5–31.
[8] M. Demirci, Fuzzy functions and their applications, J. Math. Anal. Appl. 252 (1) (2000) 495–517.
[9] D. Deutsch, Quantum theory, the Church–Turing principle and the universal quantum computer, Proc. Roy. Soc. London Ser. A 400 (1985) 97–117.
[10] D.J. Dubois, H. Prade, Fuzzy Sets and Systems: Theory and Applications, Academic Press, Inc., New York, 1980.
[11] G. Gerla, Fuzzy Logic: Mathematical Tools for Approximate Reasoning, Kluwer Academic Publishers, Dordrecht, 2001.
[12] G. Gerla, Effectiveness and multivalued logics, J. Symbolic Logic 71 (2006) 137–162.
[13] P. Di Gianantonio, Real number computation and domain theory, Inform. and Comput. 127 (1996) 11–25.
[14] D.Q. Goldin, S.A. Smolka, P.C. Attie, E.L. Sonderegger, Turing machines, transition systems and interaction, Inform. and Comput. 194 (2004) 101–128.
[15] A. Grzegorczyk, On the definitions of computable real continuous functions, Fund. Math. 44 (1957) 61–71.
[16] L. Harkleroad, Fuzzy recursion, ret's and isols, Z. Math. Logik Grundlagen Math. 30 (1984) 425–436.
[17] M.A. Harrison, Introduction to Formal Language Theory, Addison-Wesley, Reading, MA, 1978.
[18] P. Hájek, Fuzzy logic and arithmetical hierarchy, Fuzzy Sets and Systems 73 (1995) 359–363.
[19] J.E. Hopcroft, J.D. Ullman, Introduction to Automata Theory, Languages and Computation, Addison-Wesley, Reading, MA, 1979.
[20] J. Klir, B. Yuan, Fuzzy Sets and Fuzzy Logics: Theory and Applications, Prentice Hall PTR, 1995.
[21] E.T. Lee, L.A. Zadeh, Note on fuzzy languages, Inform. Sci. 4 (1) (1969) 421–434.
[22] K.H. Lee, First Course on Fuzzy Theory and Applications, Springer, Berlin, 2005.
[23] P. Linz, An Introduction to Formal Language and Automata, Jones and Bartlett Publisher, 2001.
[24] K. Menger, Statistical metrics, Proc. Nat. Acad. Sci. 37 (1942) 535–537.
[25] C. Moraga, Towards a fuzzy computability?, Mathware & Soft Comput. 6 (1999) 163–172.
[26] R. Morales-Bueno, R. Conejo, J.L. Pérez de-la Cruz, F. Triguero-Ruiz, On a class of fuzzy computable functions, Fuzzy Sets and Systems 121 (2001) 505–522.
[27] J.N. Morderson, D.S. Malik, Fuzzy Automata and Languages: Theory and Applications, Chapman & Hall, CRC, London, Boca Raton, FL, 2002.
[28] C.V. Negoita, D.A. Ralescu, Applications of Fuzzy Sets to Systems Analysis, Birkhaeuser, Basel, 1975.
[29] E. Novak, The real number model in numerical analysis, J. Complexity 11 (1) (1995) 57–73.
[30] I. Perfilieva, Fuzzy function as an approximate solution to a system of fuzzy relation equations, Fuzzy Sets and Systems 147 (2004) 363–383.

[31] E.S. Santos, Fuzzy algorithms, Inform. and Control 17 (1970) 326–339.
[32] E.S. Santos, Fuzzy and probabilistic programs, Inform. Sci. 10 (1976) 331–335.
[33] B. Schweizer, A. Sklar, Associative functions and abstract semigroups, Publ. Math. Debrecen 10 (1963) 69–81.
[34] R.I. Soare, Recursively Enumerable Sets and Degrees, Springer, Berlin, 1987.
[35] K. Weihrauch, Computable Analysis—An Introduction, Springer, Berlin, 2000.
[36] J. Wiedermann, Fuzzy Turing machines revised, Comput. Artificial Intelligence 21 (3) (2002) 1–13.
[37] J. Wiedermann, Characterizing the super-Turing computing power and efficiency of classical fuzzy Turing machines, Theoret. Comput. Sci. 317 (2004) 61–69.
[38] L.A. Zadeh, Fuzzy algorithms, Inform. and Control 2 (1968) 94–102.
[39] H.J. Zimmermann, Fuzzy Set Theory and its Applications, fourth ed., Kluwer Academic Publishers, Dordrecht, 2001.