Available online at www.sciencedirect.com
ScienceDirect
Fuzzy Sets and Systems 333 (2018) 87–105
FUZZY sets and systems
www.elsevier.com/locate/fss

Fuzzy Turing machines: Normal form and limitative theorems

Giangiacomo Gerla

Department of Mathematics University of Salerno, via Giovanni Paolo II, I-84084 Fisciano, SA, Italy

Received 21 February 2016; received in revised form 16 December 2016; accepted 18 January 2017
Available online 2 February 2017

Abstract

A normal form for fuzzy Turing machines is proposed and examined. This normal form is arithmetical in nature since the truth values are substituted by n-ples of natural numbers and the operation interpreting the conjunction becomes a sort of truncated sum. Also, some of the results in the paper enable us to emphasize the inadequacy of the notion of fuzzy Turing machine for fuzzy computability, i.e. that this notion is not a good candidate for a 'Church thesis' in the fuzzy mathematics framework.

Keywords: Fuzzy Turing machines; Fuzzy computability; Normal form; Church thesis

1. Introduction

As it is well known, the notions of '*semidecidability*', '*decidability*' and '*computability*' are basic for classical logic, computability theory, formal language theory [15]. For example, in absence of these notions almost all the results in classical formal logic lose their meaning. Indeed, a completeness theorem could be achieved simply by assuming as a system of logical axioms the whole set of logically true sentences. Also, it would be possible to obtain the axiomatizability of arithmetic by choosing as a system of proper axioms the whole set of formulas true in the canonical model. Now, exactly the same observations hold true for formal fuzzy logic and this makes crucial the definition of an adequate theory of computability in the fuzzy framework. This is particularly true for Pavelka approach to fuzzy logic [2] since it should be natural to expect from an inferential apparatus the semidecidability of the fuzzy subset of theorems of a decidable fuzzy theory.

Now, since the whole (classical) computability theory is based on the notion of Turing machine, it is not surprising that in literature definitions of fuzzy Turing machine are proposed and investigated (see [22,1,16,18,20,21]). These definitions require the use of algebraic structures to evaluate the acceptability degree of the transition rules of a fuzzy Turing machine and to calculate the truth values of the relative accepted fuzzy language.

The aim of this paper is to show that every fuzzy Turing machine whose set of truth values is totally ordered is reducible to a machine based on a numerical monoid, i.e. a totally ordered commutative monoid whose elements are (classes of) p-ples of natural numbers. Also, some of the results in the paper enable us to emphasize the inadequacy

E-mail address: gerla@unisa.it.

http://dx.doi.org/10.1016/j.fss.2017.01.008
0165-0114/

of the notion of fuzzy Turing machine for a 'Church thesis' in the fuzzy framework. All the results and considerations in the paper hold true also for the fuzzy grammars and for all the notions in fuzzy logic based on totally ordered commutative monoids (for example, fuzzy control and fuzzy logic programming).

Notice that all the theorems related to the ordered semigroups in this paper are either easy to prove or well known in specialized literature (see for example [4]). Indeed, this paper is merely an application to fuzzy logic of the powerful and elegant theory of the ordered semigroups.

2. Partially ordered monoids for the fuzzy Turing machines

We denote by N the set of natural numbers different from zero and we put

$$N_0 = N \cup \{0\}, \qquad N_\infty = N \cup \{\infty\}, \qquad N_{0,\infty} = N \cup \{0, \infty\}.$$

The usual addition and order are extended to $N_{0,\infty}$ by assuming that $\infty + x = x + \infty = \infty$ and that ∞ is the maximum. A *type* is a p-ple $(o(1), \ldots, o(p))$ of elements in N_∞. We denote by $\Re$ the set of real numbers and by $[0, 1]$ the interval $\{x \in \Re : 0 \leq x \leq 1\}$. Given an ordered set $\boldsymbol{A} = (A, \leq, 0, 1)$ in which 0 is the minimum and 1 is the maximum, we denote by $x \wedge y$ and $x \vee y$ the last upper bound and the greater upper bound of the set $\{x, y\}$ (in the case they exist). We call *fuzzy subset* of a nonempty set S a map $s : S \to A$ from S to A (for elementary notions in fuzzy set theory, see for example [9,12–14]). We put $values(s) = \{s(x) : x \in S\} - \{0, 1\}$. An *alphabet* is a finite nonempty set I, a (*formal*) *language* is a subset of the set I^+ of words on I, a *fuzzy language* is a fuzzy subset of I^+.

In fuzzy logic the domain of $\boldsymbol{A}$ is interpreted as the set of truth values and it is equipped with suitable operations to interpret all the logical connectives. Now, there are several important notions in fuzzy set theory using only an operation for the conjunction and the join operator. Examples are furnished by the fuzzy Turing machines and the fuzzy grammars. Further examples are in fuzzy control and in fuzzy logic programming. This suggests considering a particular class of complete, partially ordered monoid.

Definition 2.1. A *pomonoid* (*tomonoid*) is a structure $\boldsymbol{A} = (A, \cdot, \leq, 1)$ such that

- $(A, \cdot, 1)$ is a monoid
- $\leq$ is a partial (total) order
- the operation $\cdot$ is order-preserving, i.e. for every x, y, z in A, $x \leq y$ entails $x \cdot z \leq y \cdot z$ and $z \cdot x \leq z \cdot y$.

A structure $\boldsymbol{A} = (A, \cdot, \leq, 0, 1)$ is a *conjunction monoid* (*tomonoid*) if

- $\boldsymbol{A} = (A, \cdot, \leq, 1)$ is a commutative pomonoid (tomonoid);
- $0 \neq 1$, 0 is the minimum and 1 is the maximum in $(A, \leq)$.

Notice that, since 1 is the identity, $0 \cdot 1 = 0$ and therefore, since $\leq$ is order-preserving and 1 is the maximum, $0 \cdot x \leq 0 \cdot 1 = 0$ for every x in A. This proves that $0 \cdot x = 0$. The expression 'conjunction monoid' is suggested by the fact that in a multi-valued logic the properties listed in Definition 2.1 are the usual ones satisfied by the operation used to interpret the conjunction. We adopt the multiplicative notation since this is in accordance with the tradition in multi-valued logic, obviously there is no difficulty to adopt an additive notation.

The main examples of conjunction tomonoid are furnished by the continuous triangular norms (see for example [11]).

Definition 2.2. A *continuous triangular norm* is a continuous binary operation in the interval $[0, 1]$ which is associative, commutative and monotone and whose identity element is 1. A *T-conjunction monoid* is a conjunction monoid whose operation is a continuous triangular norm.

The following are three basic examples of a T-conjunction monoid:

- $\boldsymbol{A}$ is the *Gödel monoid* if $\cdot$ is the minimum,
- $\boldsymbol{A}$ is the *Łukasiewicz monoid* if $\cdot$ is defined by setting $x \cdot y = \max\{x + y - 1, 0\}$,

– $\boldsymbol{A}$ is the *product monoid* if $\cdot$ is the usual product.

The following definition gives an important tool to obtain continuous triangular norms.

Definition 2.3. Let I be a nonempty set, $(\otimes_i)_{i\in I}$ a family of continuous triangular norms and $(l_i, r_i)_{i\in I}$ a family of pairwise disjoint intervals contained in $[0, 1]$. Then *ordinal sum of* $(\otimes_i)_{i\in I}$ *via* $(l_i, r_i)_{i\in I}$ is the operation $\otimes$ in $[0, 1]$ defined by setting

$$x \otimes y = \begin{cases} l_i + (r_i - l_i) \cdot (\frac{x-l_i}{r_i-l_i} \otimes_i \frac{y-l_i}{r_i-l_i}) & \text{if } x, y \in [l_i, r_i] \\ \min\{x, y\} & \text{otherwise.} \end{cases} \tag{2.1}$$

It is possible to prove the following theorem.

Theorem 2.4. *The ordinal sum $\otimes$ of a family $(\otimes_i)_{i\in I}$ of triangular norms via a family $(l_i, r_i)_{i\in I}$ of pairwise disjoint intervals is a continuous triangular norm. Also, each interval $[l_i, r_i]$ is closed with respect to $\otimes$ and therefore defines a structure $([l_i, r_i], \otimes, \leq, l_i, r_i)$. This structure is isomorphic with $([0, 1], \otimes_i, \leq, 0, 1)$ via the map $f_i : [l_{i-1}, l_i] \to [0, 1]$ defined by putting $f_i(x) = \frac{x-l_{i-1}}{l_i-l_{i-1}}$.*

The importance of the three basic monoids and of the notion of ordinal sum lies in the following theorem whose proof we omit.

Theorem 2.5 *(Mostert–Shields theorem). Every continuous triangular norm is the ordinal sum of Gödel, Łukasiewicz and product triangular norms.*

Now we give some basic notions on the conjunction monoids universal in nature. Recall that the *kernel* of a map $f : A \to B$ is the equivalence relation $\equiv$ in A defined by setting $x \equiv y$ if $f(x) = f(y)$.

Definition 2.6. Given two conjunction monoids $\boldsymbol{A}^* = (A^*, \cdot, \leq, 0, 1)$ and $\boldsymbol{A} = (A, \cdot, \leq, 0, 1)$, a function $f : \boldsymbol{A}^* \to \boldsymbol{A}$ is called a *morphism* (*epimorphism*) if it is an order-preserving homomorphism (epimorphism) from $(A^*, \cdot, 0, 1)$ to $(A, \cdot, 0, 1)$. An *isomorphism* is a one-to-one morphism whose inverse is a morphism. A *congruence* is a binary relation $\equiv$ in $\boldsymbol{A}^*$ which is the kernel of a morphism f.

Observe that an one-to-one morphism f is an isomorphism if and only if, for every $x, y \in A$,

$$x \leq y \Leftrightarrow f(x) \leq f(y). \tag{2.2}$$

Notice also that in universal algebra the fact that a relation $\equiv$ is a congruence if and only if $\equiv$ is the kernel of a homomorphism is a theorem and not a definition. Indeed, the definition is given in terms of the behavior of $\equiv$ with respect to the operations. Due to the presence of an order relation, in ordered semigroup theory there are difficulties to proceed in the same way. So, we prefer to define directly a congruence as the kernel of a morphism.

Definition 2.7. Let f be a morphism from a conjunction monoid $\boldsymbol{A}^*$ to a conjunction monoid $\boldsymbol{A}$ and let $\equiv$ be the associated congruence. Then we call *quotient of* $\boldsymbol{A}^*$ *modulo* $\equiv$ the structure $\boldsymbol{A}^*/_{\equiv} = (A^*/\equiv, \cdot, \leq, [0], [1])$ where

– $(A^*/\equiv, \cdot, [0], [1])$ is the quotient of $(A^*, \cdot, 0, 1)$ modulo $\equiv$ (in the usual sense)
– $\leq$ is defined by setting $[x] \leq [y]$ if and only if $f(x) \leq f(y)$.

The *image of* $\boldsymbol{A}^*$ *by* f is the substructure $f(\boldsymbol{A})$ of $\boldsymbol{A}$ defined in the domain $f(A)$.

Observe that the equivalence classes defined by $\equiv$ are convex subsets and that $[x] \leq [y]$ if and only if either $[x] = [y]$ or every element in $[x]$ is less than of every element of $[y]$.

Proposition 2.8. *Let $f : \boldsymbol{A}^* \to \boldsymbol{A}$ be a morphism and $\equiv$ the related kernel, then*

i) *the image* $f(\boldsymbol{A}^*)$ *of* $\boldsymbol{A}^*$ *is a substructure of* $\boldsymbol{A}$ *which is a conjunction monoid*
ii) *the quotient* $\boldsymbol{A}^*/\equiv$ *is a conjunction monoid*
iii) *the map* $f_{[]}: \boldsymbol{A}^*/\equiv \to f(\boldsymbol{A})$ *defined by setting* $f_{[]}([x]) = f(x)$ *is an isomorphism.*

In particular, if f *is an epimorphism, then* $\boldsymbol{A}^*/\equiv$ *is isomorphic with* $\boldsymbol{A}$.

Proof. i) is immediate. To prove ii) we observe that $\boldsymbol{A}^*/\equiv$ is a commutative monoid in which the class [1] is the identical element and that $\leq$ is an order relation. To prove that $\leq$ is compatible with the product, assume that $[x] \leq [y]$, i.e. that $f(x) \leq f(y)$, and that $[z]$ is an element in $\boldsymbol{A}^*/\equiv$. Then, since $f(x)\cdot f(z) \leq f(y)\cdot f(z)$ and

$$f(x)\cdot f(z) \leq f(y)\cdot f(z) \Leftrightarrow f(x\cdot z) \leq f(y\cdot z) \Leftrightarrow [x\cdot z] \leq [y\cdot z] \Leftrightarrow [x]\cdot[z] \leq [y]\cdot[z],$$

the inequality $[x]\cdot[z] \leq [y]\cdot[z]$ holds true. Finally, to prove that $f_{[]}$ is an isomorphism, we observe that

$$f_{[]}\big([x]\cdot[y]\big) = f(x\cdot y) = f(x)\cdot f(y) = f_{[]}\big([x]\big)\cdot f_{[]}\big([y]\big)$$

and that $f_{[]}([0]) = f(0) = 0$, $f_{[]}([1]) = f(1) = 1$. Moreover, it is immediate that $f_{[]}$ is one-to-one and that,

$$[x] \leq [y] \Leftrightarrow f(x) \leq f(y) \Leftrightarrow f_{[]}\big([x]\big) \leq f_{[]}\big([y]\big). \qquad \square$$

3. Fuzzy Turing machines

The following definition is representative of almost all definitions of fuzzy Turing machine existing in literature (see for example [1,16,18,20–22]).

Definition 3.1. Let $\boldsymbol{A} = (A, \cdot, \leq, 0, 1)$ be a conjunction monoid such that $\leq$ is a complete order, then a *non-deterministic fuzzy Turing machine*, in brief a *FT-machine*, is a structure $\boldsymbol{\mathcal{F}}_{\boldsymbol{A}} = (Q, T, I, \Delta, q_0, q_f, \boldsymbol{A}, \mu)$ where

– Q is the finite set of *internal states*;
– T is the finite set of *tape symbols*;
– I is the set of *input symbols*, *where* $I \subseteq T$;
– $\Delta = Q \times T \times Q \times T \times \{-1, 0, +1\}$ is the set of *transition rules*;
– $\Box \in T - I$ is the *blank symbol*;
– q_0, q_f are the *initial* and the *accepting state*, respectively;
– $\mu : \Delta \to A$ is the fuzzy subset of *admitted transition rules*.

Notice that there are also alternative definitions which are based on structures different from the conjunction monoids, for examples on the semirings (see E.S. Santos in [16]). In this paper we refer only to Definition 3.1.

Then a FT-machine is like a classical non-deterministic machine but in place of the usual subset of transition rules a fuzzy subset $\mu : \Delta \to A$ of transition rules is admitted. This fuzzy subset gives the correctness degree of the admitted rules and it enables us to define the fuzzy language recognized by the fuzzy machine.

We adopt the usual definitions in Turing machine theory. So, the meaning of a transition rule $r = (q_i, t_i, q_{i+1}, t_{i+1}, d)$ is that when the machine is in state q_i and the current symbol has been read is t_i, then the machine is authorized to print t_{i+1} on the current cell, to move the scanning head according with d and to enter in the state q_{i+1}. The notions of *configuration* and of *accepting configuration* are defined as in classical theory. Given an input $w \in I^+$, we denote by $S(w)$ the configuration in which the characters of w are printed on the tape starting from the leftmost cell, the scanning head is placed atop the leftmost cell, and the machine enters state q_0. Given two configurations S_i and S_{i+1}, we write $S_i \vdash^{r(i)} S_{i+1}$ provided that $r(i)$ is a transition rule able to transform S_i into S_{i+1}. A *computational path from* S_0 to S_n is a sequence like $S_0 \vdash^{r(1)} S_1, \dots, S_{n-1} \vdash^{r(n)} S_n$. We denote by $(S_0/r(1), \dots, S_n/r(n))$ this computational path. In the case S_n is an accepting configuration and $S_0 = S(w)$, we say that $(S_0/r(1), \dots, S_n/r(n))$ is *an accepting computational path for* w. The *validity degree* of the computational path $(S_0/r(1), \dots, S_n/r(n))$ is the element $D((S_0/r(1), \dots, S_n/r(n)))$ of A defined by the equation

$$D\big((S_0/r(1), \dots, S_n/r(n))\big) = \mu(r_1)\cdot \dots \cdot \mu(r_n).$$

Definition 3.2. Given a FT-machine $\mathcal{F}$, the *fuzzy language accepted by* $\mathcal{F}$ is the fuzzy language $e : I^+ \to A$ obtained by setting, for every $w \in I^+$

$$e(w) = \sup\{D(Z) : Z \text{ is an accepting computational path for } w\}. \tag{3.1}$$

In classical computability theory the notion of language accepted by a Turing machine enables us to define the ones of semidecidability, decidability for subsets and the one of computability for a function. Indeed, a subset X of I^+ is semidecidable if and only if is accepted by a Turing machine, X is decidable if and only if both X and its complement are semidecidable, a function $f : I^+ \to I^+$ is computable if and only if is a semidecidable subset of $I^+ \times I^+$. Likewise, it should be possible to call *semidecidable* a fuzzy subset which is accepted by a fuzzy machine, *decidable* a fuzzy subset which is semidecidable together with its complement (once in $\boldsymbol{A}$ an operator is defined to interpret the negation and therefore a notion of complement is defined). Regarding the notion of computability for fuzzy functions, we have to keep in mind that a fuzzy function is defined as a fuzzy relation, i.e. a fuzzy subset of $I^+ \times I^+$. Then, again we can say that a fuzzy function is *computable* provided that it is a semidecidable fuzzy subset of $I^+ \times I^+$.

To give a representation theorem for the FT-machines, it is necessary to consider the following definition.

Definition 3.3. Let $\mathcal{F}_{\boldsymbol{A}^*} = (Q, T, I, \Delta, q_0, q_f, \boldsymbol{A}^*, \mu^*)$ and $\mathcal{F}_{\boldsymbol{A}} = (Q, T, I, \Delta, q_0, q_f, \boldsymbol{A}, \mu)$ be two FT-machines coinciding in the classical part $(Q, T, I, \Delta, q_0, q_f)$ and let $f : \boldsymbol{A}^* \to \boldsymbol{A}$ be a morphism preserving the joins. Then we say that $\mathcal{F}_{\boldsymbol{A}}$ is *reducible to* $\mathcal{F}_{\boldsymbol{A}^*}$ *via* f if, for every $x \in \Delta$,

$$\mu(x) = f\big(\mu^*(x)\big). \tag{3.2}$$

The relation of reducibility is a pre-order and, as it is usual, this pre-order is associated with an equivalence relation. Namely, we say that $\mathcal{F}_{\boldsymbol{A}^*}$ and $\mathcal{F}_{\boldsymbol{A}}$ are *equivalent* in the case $\mathcal{F}_{\boldsymbol{A}^*}$ is *reducible to* $\mathcal{F}_{\boldsymbol{A}}$ and $\mathcal{F}_{\boldsymbol{A}}$ is *reducible to* $\mathcal{F}_{\boldsymbol{A}^*}$. Trivially, in the case f is an isomorphism and (3.2) is satisfied, then the two machines are equivalent.

Theorem 3.4. *Let $\boldsymbol{\mathcal{F}}_{\boldsymbol{A}^*}$ and $\boldsymbol{\mathcal{F}}_{\boldsymbol{A}}$ be two FT-machines, and let e^*, e be the fuzzy languages accepted by these machines. Then, if $\boldsymbol{\mathcal{F}}_{\boldsymbol{A}}$ is reducible to $\boldsymbol{\mathcal{F}}_{\boldsymbol{A}^*}$ via f, we have that,*

$$e(w) = f\big(e^*(w)\big). \tag{3.3}$$

Proof. Notice that the two FT-machines have the same computational paths while they can differ in the valuation of the paths. Denote by $D((S_0/r(1), \ldots, S_n/r(n)))$ and $D^*((S_0/r(1), \ldots, S_n/r(n)))$ the degree of validity of the computational path $(S_0/r(1), \ldots, S_n/r(n))$ with respect to $\boldsymbol{\mathcal{F}}_{\boldsymbol{A}}$ and $\boldsymbol{\mathcal{F}}_{\boldsymbol{A}^*}$, respectively. Then we prove, by induction on n, that

$$D\big(\big(S_0/r(1), \ldots, S_n/r(n)\big)\big) = f\big(D^*\big(\big(S_0/r(1), \ldots, S_n/r(n)\big)\big)\big).$$

Indeed, if $n = 1$, both the sides of this equation are equal with 1. Otherwise

$$\begin{aligned} D\big(\big(S_0/r(1), \ldots, S_n/r(n)\big)\big) &= D\big(\big(S_0/r(1), \ldots, S_{n-1}/r(n-1)\big)\big) \cdot \mu(r_n) \\ &= f\big(D^*\big(\big(S_0/r(1), \ldots, S_{n-1}/r(n-1)\big)\big)\big) \cdot f\big(\mu'(r_n)\big) \\ &= f\big(D^*\big(\big(S_0/r(1), \ldots, S_{n-1}/r(n-1)\big)\big)\big) \cdot \mu'(r_n) \\ &= f\big(D^*\big(\big(S_0/r(1), \ldots, S_n/r(n)\big)\big)\big). \end{aligned}$$

Consequently,

$$\begin{aligned} e(w) &= \sup\big\{D\big(\big(S_0/r(1), \ldots, S_n/r(n)\big)\big) : S_0 = S(w),\ S_n \text{ is an accepting configuration}\big\} \\ &= \sup\big\{f\big(D^*\big(\big(S_0/r(1), \ldots, S_n/r(n)\big)\big)\big) : S_0 = S(w),\ S_n \text{ is an accepting configuration}\big\} \\ &= f\big(\sup\big\{D^*\big(\big(S_0/r(1), \ldots, S_n/r(n)\big)\big) : S_0 = S(w),\ S_n \text{ is an accepting configuration}\big\}\big) \\ &= f\big(e^*(x)\big). \qquad \square \end{aligned}$$

Let $\boldsymbol{\mathcal{F}}_{\boldsymbol{A}}$ be a FT-machine, e the fuzzy language accepted by $\boldsymbol{\mathcal{F}}_{\boldsymbol{A}}$, then in the case $\boldsymbol{\mathcal{F}}_{\boldsymbol{A}}$ is reducible to $\boldsymbol{\mathcal{F}}_{\boldsymbol{A}^*}$ via f we can calculate the value $e(w)$ in two steps:

– we calculate $e^*(w)$ by the machine $\mathcal{F}_{A^*}$,
– then we transform this value by using f.

This is useful in the case the computation in $\boldsymbol{A}^*$ is simpler than the one in $\boldsymbol{A}$, obviously.

4. Finitely generated and complete conjunction monoids

As it is usual, the notion of n-power of an element x of a complete monoid $\boldsymbol{A}$ is defined by the equations $x^0 = 1$, $x^{n+1} = x^n \cdot x$. In the case the order in $\boldsymbol{A}$ is complete, we put $x^\infty = \inf\{x^n : n \in N\}$, too.

Proposition 4.1. *Let $\boldsymbol{A}$ be a conjunction monoid, then, for every $x_1, \dots, x_n$ in A,*

$$x_1 \cdot \ldots \cdot x_n \leq x_1, \dots, x_1 \cdot \ldots \cdot x_n \leq x_n. \tag{4.1}$$

Also, for every $x \in A$, the sequence $(x^n)_{n\in N}$ is order-reversing. Namely, either $(x^n)_{n\in N}$ is an infinite strictly order-reversing sequence, i.e. $x^1 > x^2 > \ldots x^n > \ldots$ or there is $m \in N$ such that $x^1 > x^2 > \ldots > x^m = x^{m+1} = x^{m+2} = \ldots$, i.e. the sequence stops after a finite number of steps.

Proof. At first, observe that for every x and y since $x \leq 1$, $x \cdot y \leq 1 \cdot y = y$. Likewise one proves that $x \cdot y \leq x$. Now to prove (4.1) we observe that, in the case $n = 1$, the inequality is trivial. Assume that (4.1) is satisfied for $n = i$. Then from the inequality $x_1 \cdot \ldots \cdot x_i \leq x_1$, we have that $x_1 \cdot \ldots \cdot x_i \cdot x_{i+1} \leq x_1.x_{i+1} \leq x_1$ and from the inequality $x_1 \cdot \ldots \cdot x_i \leq x_i$ we obtain that $x_1 \cdot \ldots \cdot x_i \cdot x_{i+1} \leq x_i \cdot x_{i+1} \leq x_{i+1}$. So (4.1) holds true in the case $n = i + 1$.

To prove the second part of the proposition, we have to prove that $x^n \geq x^{n+1}$ for every natural number n. This is trivial since $x^1 = x \geq x^2$ and $x^i \geq x^{i+1}$ entails that $x^{i+1} = x^i \cdot x \geq x^{i+1} \cdot x = x^{i+1}$. The remaining part of the proposition is equally trivial. □

Definition 4.2. Let G be a subset of a conjunction monoid $\boldsymbol{A}$ and denote by $<G>$ the conjunction monoid obtained by intersecting all the substructures of $\boldsymbol{A}$ containing G. If $<G> = A$, then we say that G is a *system of generators* of $\boldsymbol{A}$ and that $\boldsymbol{A}$ is *generated* by G. We say that an element x *depends on the elements of* G provided that $x \in <G>$. If G is finite, we say that $\boldsymbol{A}$ is *finitely generated*.

Trivially, x depends on G if and only if either $x \in \{0, 1\}$ or x is a product of elements in G. In accordance with Proposition 4.1, in this case x is product of elements in G these elements are greater or equal with x.

Proposition 4.3. *Let $f : A^* \to A$ be an epimorphism from a conjunction monoid $\boldsymbol{A}^*$ into a conjunction monoid $\boldsymbol{A}$ and assume that G is a system of generators for $\boldsymbol{A}^*$. Then $f(G)$ is a system of generators for $\boldsymbol{A}$. Consequently, the epimorphic image and the quotient of a finitely generated conjunction monoid is a finitely generated conjunction monoid.*

Proof. Let y be an element in $A - \{0, 1\}$ and let $x \in A^*$ be such that $f(x) = y$. Then $x \neq 0$ and $y \neq 1$ and therefore there are $g_1, \dots, g_n \in B$ such that $x = g_1 \cdot \ldots \cdot g_n$. Then, $y = f(x) = f(g_1 \cdot \ldots \cdot g_n) = f(g_1) \cdot \ldots \cdot f(g_n)$. □

The next propositions play a crucial role in this paper.

Proposition 4.4. *Let $\boldsymbol{A} = (A, \cdot, \leq, 0, 1)$ be a finitely generated conjunction monoid, then*

i) *every sequence of elements of A contains a subsequence which is order-reversing;*
ii) *there is no strictly increasing sequence of elements in A;*
iii) *every nonempty subset of A admits a maximal element.*

Proof. To prove i), let $(s_n)_{n\in N}$ be a sequence of elements of A, then we distinguish the following cases.

Case 1. Assume that the set $\{n \in N : s_n = 0\}$ is empty. Then, given a system of generators $g_1, \dots, g_p$ of $\boldsymbol{A}$, for every $n \in N$ there is $(f(1,n), \dots, f(p,n)) \in N_0^p$ such that $s_n = g_1^{f(1,n)} \cdot \ldots \cdot g_p^{f(p,n)}$. Now, it is evident that every sequence $(a(n))_{n\in N}$ of natural numbers admits an order-preserving subsequence $(a(h(n)))_{n\in N}$. Indeed

– if $(a(n))_{n\in N}$ is bounded, then there is $\underline{n} \in N_0$ such that $\{n \in N : a(n) = \underline{n}\}$ is infinite and therefore an order-preserving sequence exists,
– if $(a(n))_{n\in N}$ is not bounded, then there is a strictly increasing subsequence of $(a(n))_{n\in N}$.

In accordance with this fact,

– the sequence $(f(1,n))_{n\in N}$ admits a order-preserving subsequence $(f(1,h(1,n)))_{n\in N}$,
– the sequence $(f(2,h(1,n)))_{n\in N}$ admits an order-preserving subsequence $(f(2,h(2,n)))_{n\in N}$,
– ...
– the sequence $(f(p,h(p-1,n)))_{n\in N}$ admits an order-preserving subsequence $(f(p,h_p(p,n)))_{n\in N}$.

It is evident that $(f(1,h(1,n)), f(2,h(2,n)), \ldots, f(p,h(p,n)))_{n\in N}$ is an order-preserving subsequence of $((f(1,n), \ldots, f(p,n)))_{n\in N}$ with respect to the pointwise order in N_0^p and therefore that $(g_1^{h(1,n)} \cdot g_2^{h(2,n)} \cdot \ldots \cdot g_p^{h(p,n)})_{n\in N}$ is an order-reversing subsequence of $(s_n)_{n\in N}$.

Case 2. The set $\{n \in N : s_n = 0\}$ is finite. Then, if m is the maximum of this set, the sequence $(s_{n+m+1})_{n\in N}$ is a subsequence of $(s_n)_{n\in N}$ satisfying the condition of Case 1. This entails that $(s_{n+m+1})_{n\in N}$ admits an order-reversing subsequence.

Case 3. The set $\{n \in N : s_n = 0\}$ is infinite. Then, it is evident that we can consider a subsequence of $(s_n)_{n\in N}$ constantly equal to 0.

Claim ii) is a trivial consequence of i), to prove iii) let S be a nonempty set and assume that in S there is no maximal element. Then, given $\underline{s} \in S$, we can define the sequence $(s_n)_{n\in N}$ by induction by putting $s_1 = s$ and s_{n+1} equal to any element of S such that $s_{n+1} > s_n$. Then $(s_n)_{n\in N}$ is a strictly increasing sequence of elements of S in contrast with ii). □

In the next proposition, we denote by $\leq_d$ the *dual* of an order relation $\leq$, i.e. the relation defined by setting

$$x \leq_d y \Leftrightarrow y \leq x.$$

Proposition 4.5. *Assume that $\boldsymbol{A}$ is a conjunction tomonoid and that $\boldsymbol{A}^*$ is a finitely generated substructure of $\boldsymbol{A}$, then every nonempty subset X of $\boldsymbol{A}^*$ admits a maximum. This means that $\leq_d$ is a well order in A^*. This well order is enumerable, bounded and therefore complete. Moreover, the join$_{\boldsymbol{A}}$ of a subset X in $\boldsymbol{A}$ coincides with the join$_{\boldsymbol{A}^*}$ of X in $\boldsymbol{A}^*$ and therefore with the maximum of X.*

Proof. Let X be nonempty subset X of A^*. Then, by iii) of Proposition 4.5, X admits a maximal element. Since the order in $\boldsymbol{A}^*$ is total, this element is a maximum. This means that $\leq_d$ is a well order. It is evident that 0 is the maximum and 1 is the minimum and that $\leq_d$ is complete. For example, observe that since the set $\{x \in A$: such that x is an upper bound of $X\}$ contains 0, this set is nonempty and therefore it admits a maximum. By definition, this maximum is the meet of X in $\boldsymbol{A}^*$.

Let $join_{\boldsymbol{A}}(X)$ be the join of X in $\boldsymbol{A}$, then $join_{\boldsymbol{A}}(X) \geq x$ for every $x \in X$ and therefore since $\max(X) \in X$, $join_{\boldsymbol{A}}(X) \geq \max(X)$. On the other hand since $\max(X)$ is an upper bound for X, $join_{\boldsymbol{A}}(X) \leq \max(X)$. Thus $join_{\boldsymbol{A}}(X) = \max(X) = join_{\boldsymbol{A}^*}(X)$. □

5. Basis, dimension, type, ordinal of a conjunction monoid

We can give a suitable notion of basis of a finitely generated conjunction monoid.

Definition 5.1. A *basis* for a conjunction monoid $\boldsymbol{A}$ is a *minimal* system of generators, i.e. a system of generators G of $\boldsymbol{A}$ such that no proper subset of G is able to generate $\boldsymbol{A}$.

The proof of the following proposition is trivial.

Proposition 5.2. *A subset G of A is a basis for $\boldsymbol{A}$, if and only if*

– *0 and 1 are not in G;*
– *every element x in $A - \{0, 1\}$ is a product of elements in G;*
– *no element g in G depends on the remaining elements, i.e. is a product of elements in $G - \{g\}$.*

Proposition 5.3. *Every finitely generated conjunction monoid $\boldsymbol{A}$ admits a basis. Assume that $\boldsymbol{A}$ is a tomonoid, then, in the case $\boldsymbol{A}$ is the two element tomonoid its basis is $\varnothing$. Otherwise the basis is the set $\{b_1,\ldots, b_p\}$ of element with, $b_1 < \ldots < b_p$, defined by setting*

$$b_p = \max\big(A - \{<\varnothing>\}\big);$$
$$b_{p-1} = \max\big(A - <\{b_p\}>\big) \textit{ provided that } A - <\{b_p\}> \neq \varnothing;$$
$$\ldots$$
$$b_{p-2} = \max\big(A - <\{b_{p-1}, b_p\}>\big) \textit{ provided that } A - <\{b_{p-1}, b_p\}> \neq \varnothing;$$
$$b_1 = \max\big(A - <\{b_2, \ldots, b_p\}>\big) \textit{ provided that } A - <\{b_2, \ldots, b_p\}> \neq \varnothing \textit{ and } A = <\{b_1, \ldots, b_p\}>.$$

Proof. It is evident that the basis of the two elements conjunction monoid is $\varnothing$. Otherwise the set

$$\{n \in N\text{: there is a system of } n \text{ generators for } \boldsymbol{A}\}$$

is nonempty and we can consider the natural number

$$p = \min\{n\text{: there is a system of } n \text{ generators for } \boldsymbol{A}\}.$$

Trivially, a system of p generators is a basis. Denote by $b_1 < \ldots < b_p$ this basis. To prove that $b_p = \max(A - \{0, 1\})$ it is sufficient to observe that if x is any element different from 0 and 1 and $x = b_1^{n(1)} \cdot \ldots \cdot b_p^{n(p)}$, then there is $i \in \{1, \ldots, p\}$ such that $n(i) \neq 0$. Also, we have that

$$x = b_1^{n(1)} \cdot \ldots \cdot b_p^{n(p)} \leq b_1 \wedge \ldots \wedge b_p \leq b_i \leq b_p.$$

Since $b_p \in A - \{1\}$, this proves that b_p is the maximum of $A - \{1\}$. To prove that $b_{p-1} = \max(A - <b_p>)$, assume that $x \in A - <b_p>$ and $x = b_1^{n(1)} \cdot \ldots \cdot b_p^{n(p)}$, then there is $i \in \{1, \ldots, p-1\}$ such that $n(i) \neq 0$. This entails that $x \leq b_i \leq b_{p-1}$ and therefore, since $b_{p-1} \in A - <b_p>$, we have that $b_{p-1} = \max(A - <b_p>)$. In general we can prove that, for $i = 0, \ldots, p-1$, $b_{p-i} = \max(A - <b_{p-i+1}, \ldots, b_p>)$. Indeed, if $x \notin <b_{p-i+1}, \ldots, b_p>$ and $x = b_1^{n(1)} \cdot \ldots \cdot b_p^{n(p)}$, then there is a suitable $i \in \{1, \ldots, p-i\}$ such that $n(i) \neq 0$. Then $x \leq b_i \leq b_{p-I}$ and this shows that $b_{p-i} = \max(A - <b_{p-i} \cdot \ldots \cdot b_p>)$. □

Corollary 5.4. *The basis of a finitely generated conjunction monoid is unique.*

Proof. We observe that the system of equations in Proposition 5.3 univocally identifies the elements of a basis. This entails that this basis is unique. □

The fact that the basis is unique enables us to give the following definitions where *order*(x) is defined by setting

$$order(x) = \begin{cases} \min\{n \in N : x^n = x^{n+1}\} & \text{if the set } \{n \in N : x^n = x^{n+1}\} \text{ is nonempty} \\ \infty & \text{otherwise,} \end{cases}$$

and where we write $[\![a(1), \ldots, a(n)]\!]$ to denote a sequence of elements such that $a(1) < \cdots < a(n)$.

Definition 5.5. Let $\boldsymbol{A}$ be a finitely generated conjunction tomonoid and $[\![b_1, \ldots, b_p]\!]$ its basis, then we call

– *dimension of* $\boldsymbol{A}$ the number p and we denote it by $dim(\boldsymbol{A})$
– *type of* $\boldsymbol{A}$ the p-ple $(ord(b_1), \ldots, ord(b_p))$ and we denote it by $type(\boldsymbol{A})$.
– *ordinal of* $\boldsymbol{A}$ the ordinal number of $\leq_d$ and we denote it by $ordinal(\boldsymbol{A})$.

The proof of this proposition is trivial.

Proposition 5.6. *Let $\boldsymbol{A}$ and $\boldsymbol{A}^*$ be two conjunction monoids and $f : \boldsymbol{A}^* \to \boldsymbol{A}$ an epimorphism. Then $dim(\boldsymbol{A}^*) \leq dim(\boldsymbol{A})$. If f is an isomorphism and $\{b_1, \dots, b_p\}$ is the basis of $\boldsymbol{A}^*$, then $\{f(b_1), \dots, f(b_p)\}$ is the basis of $\boldsymbol{A}$. Therefore, $dim(\boldsymbol{A}) = dim(\boldsymbol{A}^*)$ and $tipe(\boldsymbol{A}) = type(\boldsymbol{A}^*)$. Obviously, if $\boldsymbol{A}$ is a tomonoid, then $ordinal(\boldsymbol{A}) = ordinal(\boldsymbol{A}^*)$, too.*

We conclude this section by giving a simple algorithm to calculate the basis of a finitely generated conjunction tomonoid by starting from a system of generators.

Definition 5.7. Given a sequence $[\![x(1), \dots, x(n)]\!]$ of elements in a conjunction tomonoid $\boldsymbol{A}$, and $i \in N$, we define the operator H_i by setting

$$H_i\big([\![x(1), \dots, x(n)]\!]\big) = \begin{cases} [\![x(1), \dots, x(n)]\!] \text{ in the case } i \geq n \\ \text{the sequence obtained from } [\![x(1), \dots, x(n)]\!] \text{ by deleting all the elements of} \\ [\![x(1), \dots, x(i)]\!] \text{ depending on } \{x(i+1), \dots, x(n)\}. \end{cases}$$

If $[\![a(1), \dots, a(n)]\!]$ is a system of generators, then the set

$$B = H_1\big(\dots\big(H_{n-1}\big([\![a(1), \dots, a(n)]\!]\big)\big)\dots\big)$$

is the basis of $\boldsymbol{A}$ contained in $[\![a(1), \dots, a(n)]\!]$.

Proof. Given two sequences $[\![a(1), \dots, a(n)]\!]$ and $[\![b(1), \dots, b(m)]\!]$ we say that $[\![a(1), \dots, a(n)]\!]$ is *contained* in $[\![b(1), \dots, b(m)]\!]$, and we write $[\![a(1), \dots, a(n)]\!] \subseteq [\![b(1), \dots, b(m)]\!]$ if every element in the first sequence is an element of the second sequence. Then it is immediate that $H_i([\![a(1), \dots, a(n)]\!]) \subseteq H_{i+1}([\![a(1), \dots, a(n)]\!])$ and that $<H_i([\![a(1), \dots, a(n)]\!])> = <H_{i+1}([\![a(1), \dots, a(n)]\!])> = \boldsymbol{A}$. This proves that B is a system of generators of $\boldsymbol{A}$ contained in B. To prove that B is a basis, we prove that no element in B depends from the remaining ones. To this aim, it is sufficient to observe that

- the last element in $[\![a(1), \dots, a(n)]\!]$ does not depends from the remaining ones
- the last two elements in $H_{n-1}([\![a(1), \dots, a(n)]\!])$ does not depend from the remaining ones
- the last three elements in $H_{n-2}([\![a(1), \dots, a(n)]\!])$ does not depend from the remaining ones
- ... □

For example, if we consider the usual product in the interval [0, 1], then to calculate $H_1(H_2(H_3([\![35, 0.49, 0.5, 0.7]\!])))$ we proceed as follows

1. $H_3([\![0.35, 0.49, 0.5, 0.7]\!]) = [\![0.35, 0.5, 0.7]\!]$ (since 0.49 is the unique element in $[\![0.35, 0.49, 0.5]\!]$ which is a power of 0.7).
2. $H_2([\![0.35, 0.5, 0.7]\!]) = [\![0.5, 0.7]\!]$ (since 0.35 is the unique element in $[\![0.35, 0.49]\!]$ which is a product of elements in $[\![0.5, 0.7]\!]$.
3. $H_2([\![0.5, 0.7]\!]) = [\![0.5, 0.7]\!]$, we observe that 0.35 is the unique element in $[\![0.35, 0.49]\!]$ which is a product of elements in $[\![0.5, 0.7]\!]$. By deleting this number we obtain
4. $H_1([\![0.5, 0.7]\!]) = [\![0.5, 0.7]\!]$.

This means that $[\![0.5, 0.7]\!]$ is the basis.

6. A normal form for the fuzzy Turing machines based on a tomonoid

This paper focuses its attention only to fuzzy Turing machines based on a tomonoid.

Definition 6.1. We call *FTT-machine*, a FT-machine based on a conjunction tomonoid.

The following theorem shows that in considering a FTT-machine, it is not restrictive to assume that this machine is based on a finitely generated conjunction tomonoid.

Theorem 6.2. *Every FTT-machine $\mathcal{F}_{\boldsymbol{A}}$ based on a tomonoid $\boldsymbol{A}$ is reducible to a FTT-machine $\mathcal{F}_{\boldsymbol{A}^*}$ defining the same language than $\mathcal{F}_{\boldsymbol{A}}$ and in which A^* is the substructure of A generated by values(μ) and therefore it is a finitely generated conjunction tomonoid.*

Proof. Assume that $\mathcal{F}_{\boldsymbol{A}} = (Q, T, I, \Delta, q_0, q_f, \boldsymbol{A}, \mu)$ and that $\boldsymbol{A}^*$ the substructure of $\boldsymbol{A}$ generated by *values*(μ). Then, since *values*(μ) is a finite set, $\boldsymbol{A}^*$ is a complete conjunction tomonoid and we are authorized to consider the machine $\mathcal{F}_{\boldsymbol{A}^*} = (Q, T, I, \Delta, q_0, q_f, \boldsymbol{A}^*, \mu)$ obtained from $\mathcal{F}_{\boldsymbol{A}}$ by substituting $\boldsymbol{A}$ with $\boldsymbol{A}^*$. Trivially, $\mathcal{F}_{\boldsymbol{A}}$ is reducible to $\mathcal{F}_{\boldsymbol{A}^*}$ via the identity embedding. □

In accordance with this theorem, in the sequel we assume that in all the considered FTT-machines the tomonoid is finitely generated.

Proposition 6.3. *Given an element $o \in N_\infty$, define the structure $([0, o], +^o, \leq^o, o, 0)$ where*

- $[0, o] = \{x \in N_{0,\infty} : x \leq o\}$
- $+^o$ *is the o-truncated sum, i.e.* $x +^o y = (x + y) \wedge o$
- $\leq^o$ *is the dual of the usual order* $\leq$ *in* $N_{0,\infty}$.

Then this structure is a total ordered conjunction monoid.

Proof. Trivial. □

Notice that in $([0, o], +^o, \leq^o, o, 0)$ o is the minimum and 0 is the maximum.

Definition 6.4. Given a type $\underline{o} = (o(1), \ldots, o(p))$ denote by $F^{\underline{o}}$ the set $[0, o(1)] \times \cdots \times [0, o(p)] \cup \{\underline{\infty}\}$ where $\underline{\infty}$ is not an element of $[0, o(1)] \times \cdots \times [0, o(p)]$. Then we define the structure

$$\boldsymbol{F}^{\underline{o}} = \left(F^{\underline{o}}, +^{\underline{o}}, \leq^{\underline{o}}, \underline{\infty}, 0_p\right) \tag{6.1}$$

where 0_p is the p-ple whose elements are all equal to 0 and $+^o$ and $\leq^o$ are defined by the conditions,

$$x +^{\underline{o}} \underline{\infty} = \underline{\infty} +^{\underline{o}} x = \underline{\infty} \quad \text{and} \quad \underline{\infty} \leq^{\underline{o}} x, \text{ for every } x \in F^{\underline{o}}$$
$$(n_1, \ldots, n_p) +^{\underline{o}} (m_1, \ldots, m_p) = \left(n_1 +^{o(1)} m_1, \ldots, n_p +^{o(p)} m_p\right)$$
$$(n_1, \ldots, n_p) \leq^{\underline{o}} (m_1, \ldots, m_p) \Leftrightarrow n_1 \leq^{o(1)} m_1, \ldots, m_p \leq^{o(p)} m_p.$$

Proposition 6.5. *Given a type $\underline{o} = (o(1), \ldots, o(p))$, the structure $\boldsymbol{F}^{\underline{o}}$ is a complete conjunction monoid whose basis is the set*

$$\left\{(1, 0, \ldots, 0), (0, 1, 0, \ldots, 0), \ldots, (0, \ldots, 1)\right\}$$

of p-vectors.

Proof. We confine ourselves to prove the completeness property. Let X be a nonempty subset of $F^{\underline{o}}$, then in the case $\underline{\infty} \in X$, we have that $\sup(X) = \underline{\infty}$. In the case there is $i \leq p$ such that the set

$$pr_i(X) = \left\{n \in N_0 : (n_1, \ldots, n_{i-1}, n, n_{i+1}, \ldots, n_p) \in X\right\}$$

is infinite, then $\sup(X) = \underline{\infty}$, too. Finally, in the case all the sets $pr_i(X)$ are finite, we have that $\sup(X) = (\max(pr_1(X)), \ldots, \max(pr_p(X)))$. □

Definition 6.6. Given a type $\underline{o} = (o(1), \ldots, o(p))$, we call *numerical conjunction monoid* every complete conjunction monoid which is a quotient of $\boldsymbol{F}^{\underline{o}}$. We call *numerical FT-machine* a *FT*-machine on a numerical conjunction monoid.

Proposition 6.7. *Let $\boldsymbol{A} = (A, \cdot, \leq, 0, 1)$ be a conjunction monoid with basis $[\![b(1), \dots, b(p)]\!]$ and type $\underline{o} = (o(1), \dots, o(p))$, and define the map $h : F^{\underline{o}} \to A$ by setting*

$$h(x) = \begin{cases} 0 & \text{if } x = \underline{\infty}, \\ b_1^{n(1)} \cdot \dots \cdot b_p^{n(p)} & \text{if } x = (n(1), \dots, n(p)). \end{cases} \tag{6.2}$$

Then h is an epimorphism from $\boldsymbol{F}^{\underline{o}}$ to $\boldsymbol{A}$, we call canonical epimorphism.

Proof. Trivially $h(0_p) = 1$ and, by definition, $h(\underline{\infty}) = 0$. Also, for every $x \in F^{\underline{o}}$,

$$h\big(x +^{\underline{o}} \underline{\infty}\big) = h(\underline{\infty}) = 0 = h(x) \cdot 0 = h(x) \cdot h(\underline{\infty})$$

and

$$\begin{aligned} h\big(\big(n(1), \dots, n(p)\big) +^{\underline{o}} \big(m(1), \dots, m(p)\big)\big) &= h\big(\big(n(1) + m(1), \dots, n(p) + m(p)\big)\big) \\ &= a_1^{n(1)+m(1)} \cdot \dots \cdot a_p^{n(p)+m(p)} \\ &= a_1^{n(1)} \cdot \dots \cdot a_p^{n(p)} \cdot a_1^{m(1)} \cdot \dots \cdot a_p^{m(p)} \\ &= h\big(\big(n(1), \dots, n(p)\big)\big) \cdot h\big(\big(m(1), \dots, m(p)\big)\big). \end{aligned}$$

To prove that h is order-preserving, it is sufficient to observe that

$$\begin{aligned} \big(n(1), \dots, n(p)\big) \leq^{\underline{o}} (m_1, \dots, m_p) &\Leftrightarrow m_1 \leq n_1, \dots, m_p \leq n_p \\ &\Rightarrow a_1^{n(1)} \cdot \dots \cdot a_p^{n(p)} \leq a_1^{m(1)} \cdot \dots \cdot a_p^{m(p)} \\ &\Leftrightarrow h\big(\big(n(1), \dots, n(p)\big)\big) \leq h\big((m_1, \dots, m_p)\big). \qquad \square \end{aligned}$$

Theorem 6.8. *Let $\boldsymbol{A} = (A, \cdot, \leq, 0, 1)$ be a complete, finitely generated conjunction tomonoid, then $\boldsymbol{A}$ is isomorphic with the quotient $\boldsymbol{F}^{\underline{o}}/\equiv$ where $\underline{o} = (o(1), \dots, o(p))$ is the type of $\boldsymbol{A}$ and $\equiv$ is the kernel of the canonical epimorphism. This means that every FTT-machine is equivalent with a numerical FTT-machine.*

From a computational point of view perhaps it is more convenient to consider a concrete representation of the quotient in which the elements we have to manipulate are p-ples and not classes of *p-ples*. To obtain this, it is sufficient to apply a procedure which is usual in mathematics and which is on the basis of the rewriting systems (see for example the language *Mathematica*). The idea is to represent each class of a quotient by a particular element in the class. In our case we consider in $F^{\underline{o}}$ a well order. For example we can consider the order $\leq_L$ coinciding with the lexicographic order in $[0, o(1)] \times \dots \times [0, o(p)]$ and in which $\underline{\infty}$ is the maximum. This enables us to define a choice function $c : F^{\underline{o}}/\equiv \to F^{\underline{o}}$ by setting

$$c\big([x]\big) = \min_L\big([x]\big)$$

where $\min_L$ denotes the minimum with respect to $\leq_L$. Also, we define the structure $\boldsymbol{D} = (D, \oplus, \preccurlyeq, \underline{\infty}, 0_p)$ by setting

- $D = \{c([x]) : x \in F^{\underline{o}}\}$
- $x \oplus y = c([x +^{\underline{o}} y])$
- $x \preccurlyeq y$ if and only if $h(x) \leq h(y)$.

It is immediate that the map $c : F^{\underline{o}}/_\equiv \to D$ is an isomorphism from $\boldsymbol{F}^{\underline{o}}/_\equiv$ to $\boldsymbol{D}$ and therefore that the map $k : A \to D$ defined by setting

$$k(x) = \begin{cases} \underline{\infty} & \text{if } x = 0 \\ \min_L\{(n_1, \dots, n_p) \in F^{\underline{o}} : b_1^{n(1)} \cdot \dots \cdot b_p^{n(p)} = x\} & \text{if } x \neq 0 \end{cases} \tag{6.3}$$

is an isomorphism from $\boldsymbol{A}$ to $(D, \oplus, \preccurlyeq, \underline{\infty}, 0_p)$. This proves that $\boldsymbol{D}$ is a complete, finitely generated conjunction tomonoid.

Definition 6.9. We call *ground conjunction monoid* a complete finitely generated conjunction tomonoid whose elements are either p-ples of elements in $N_{0,\infty}$ or $\underline{\infty}$. We call *ground FTT-machine* a FTT-machine which is based on a ground conjunction monoid.

Theorem 6.10. *Every total FTT-machine is equivalent with a ground FTT-machine.*

Proof. Given a FTT-machine $\mathcal{F}_{\boldsymbol{A}}$ let $k : A \to D$ the isomorphism defined by (6.3) and denote by $\mathcal{F}_{\boldsymbol{D}}$ the FT-machine obtained by substituting $\boldsymbol{A}$ with $\boldsymbol{D}$ and μ with the fuzzy subset $\mu^* : \Delta \to D$ defined by setting $\mu^*(x) = k^{-1}(\mu(x))$. Then, trivially,

$$\mu(x) = k\big(\mu^*(x)\big) \tag{6.4}$$

and this proves that $\mathcal{F}_{\boldsymbol{A}}$ is reducible to $\mathcal{F}_{\boldsymbol{D}}$. It is also evident that $\mathcal{F}_{\boldsymbol{D}}$ is reducible to $\mathcal{F}_{\boldsymbol{A}}$. □

7. Some concrete examples

In this section we show some concrete examples of applications of the proved theorems. We consider FTT-machines based on a triangular norm. Namely, we call *Zadeh machine* a machine based on a finitely generated substructure of the Gödel monoid. In a similar way we define the *Łukasiewicz machines*, the *product machines and* the *T-machines*.

7.1. Zadeh machines

The proof of the following proposition is immediate.

Proposition 7.1. *In a Zadeh machine we have that*

- *order*$(x) = 1$ *for every* $x \in A$,
- *the basis of* $\boldsymbol{A}$ *coincides with values*$(\mu) = [\![b_1, \dots, b_p]\!]$,
- *dim*$(\boldsymbol{A}) = p$,
- *type*$(\boldsymbol{A}) = (1, \dots, 1)$,
- *ordinal*$(\boldsymbol{A}) = p + 2$.

In accordance with this proposition, it is possible to simulate a Zadeh machine by a digital machine. Indeed, if h is the canonical epimorphism, the elements of $\boldsymbol{F}^{(1,\dots,1)}/\equiv$ are

$$h^{-1}(0) = \{\underline{\infty}\}$$
$$h^{-1}(1) = \{0_p\}$$

and, for x different from 0 and 1 and $i = 1, \dots, p$,

$$h^{-1}(b_i) = \big\{(0, \dots, 0, 1, \dots, 1)\big\},\ (i - 1 \text{ consecutive 0 followed by } p - i + 1 \text{ consecutive 1}).$$

Since every equivalence class has only one element, the function c is univocally defined by setting $c(\{x\}) = x$. So, $\boldsymbol{A}$ is isomorphic with the ground conjunction tomonoid via the map associating

0 with $\underline{\infty}$;

b_1 with $(1, 1, \dots, 1)$

b_2 with $(0, 1, 1, \dots, 1)$

...

b_p with $(0, 0, \dots .0, 1)$

1 with $(0, 0, \dots .0, 0)$.

In this set D of p-ple, $\oplus$ coincides with the join operation and $\preccurlyeq$ is the dual of the lexicographic order plus the condition that $\underline{\infty}$ is the minimum. This representation shows that two Gödel submonoids with the same dimension (equivalently, the same cardinality) are isomorphic. Namely, by assuming that the basis of these monoids are $[\![b_1, \dots, b_p]\!]$ and $[\![\underline{b}_1, \dots, \underline{b}_p]\!]$, we obtain an isomorphism f by setting $f(0) = 0$, $f(1) = 1$, $f(b_i) = \underline{b}_i$.

7.2. Łukasiewicz machines

The conjunction monoid of a Łukasiewicz machine satisfies the following proposition.

Proposition 7.2. *In a Łukasiewicz machine we have that, for every $x \in A - \{1\}$,*

$$order(x) = \min\big\{n : n \geq 1/(1-x)\big\}. \tag{7.1}$$

Moreover, $\boldsymbol{A}$ is finite and therefore $ordinal(\boldsymbol{A})$ is finite.

Proof. It is easy to prove that

$$x^n = \max\{n \cdot x - n + 1, 0\} \tag{7.2}$$

and therefore that $x^n = 0$ if $n \cdot x - n + 1 \leq 0$ and $x^n = n \cdot x - n + 1 > 0$ otherwise. Equivalently, $x^n = 0$ if $n \geq 1/(1-x)$ and $x^n = n \cdot x - n + 1 > 0$, otherwise. Then if $q = \min\{n : n \geq 1/(1-x)\}$ we have that $x^1 > \dots > x^q = x^{q+1}$ and this proves (7.1)

Since $ord(x)$ is finite for every $x \in A$ and $\boldsymbol{A}$ is finitely generated, $\boldsymbol{A}$ is finite. □

Notice that the cardinality of $\boldsymbol{A}$, and therefore the finite ordinal of $\boldsymbol{A}$, depends on the distance of the generators from 0. For example, assume that $\boldsymbol{A}$ is generated by 0.5, then $ord(0.5) = 2$, $A = \{0, 0.5, 1\}$ and therefore $ordinal(\boldsymbol{A})$. Assume that $\boldsymbol{A}$ is generated by 0.9, then $A = \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$ and therefore $ordinal(\boldsymbol{A}) = 11$.

To examine a case in which the dimension is different from 1, assume that $values(\mu) = \{0.1, 0.4, 0.5, 0.6, 0.7\}$. Then, in accordance with Definition 5.7,

- $H_4([\![0.1, 0.4, 0.5, 0.6, 0.7]\!]\} = [0.5, 0.6, 0.7]$ (since 0.1 and 0.4 are in $\{0.7^2, 0.7^3\}$)
- $H_3([\![0.5, 0.6, 0.7]\!]) = [\![0.5, 0.6, 0.7]\!]$
- $H_2([\![0.5, 0.6, 0.7]\!]) = [\![0.5, 0.6, 0.7]\!]$
- $H_1([\![0.5, 0.6, 0.7]\!]) = [\![0.5, 0.6, 0.7]\!]$.

This means that the basis is $[\![0.5, 0.6, 0.7]\!]$ and the type $(1, 2, 3)$. Moreover, $A = \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 1\}$ and therefore $ordinal(\boldsymbol{A}) = 9$. Regarding the ground numerical representation of $\boldsymbol{A}$, we have that the congruence classes are

$$
\begin{aligned}
&k(1) = c\big(h^{-1}(1)\big) = c\big(\big\{(0,0,0)\big\}\big) = (0,0,0)\\
&k(0.7) = c\big(h^{-1}(0.7)\big) = c\big(\big\{(0,0,1)\big\}\big) = (0,0,1)\\
&k(0.6) = c\big(h^{-1}(0.6)\big) = c\big(\big\{(0,1,0)\big\}\big) = (0,1,0)\\
&k(0.5) = c\big(h^{-1}(0.5)\big) = c\big(\big\{(1,0,0)\big\}\big) = (1,0,0)\\
&k(0.4) = c\big(h^{-1}(0.4)\big) = c\big(\big\{(0,0,2)\big\}\big) = (0,0,2)\\
&k(0.3) = c\big(h^{-1}(0.3)\big) = c\big(\big\{(0,1,1)\big\}\big) = (0,1,1)\\
&k(0.2) = c\big(h^{-1}(0.2)\big) = c\big(\big\{(0,2,0),(1,0,1)\big\}\big) = (0,2,0)\\
&k(0.1) = c\big(h^{-1}(0.1)\big) = c\big(\big\{(0,0,3),(1,1,0)\big\}\big) = (0,0,3)\\
&k(0) = \underline{\infty}.
\end{aligned}
$$

So, we have that

$$D = \{\underline{\infty}, (0,0,3), (0,2,0), (0,1,1), (0,0,2), (1,0,0), (0,1,0)(0,0,1), (0,0,0)\}$$

where the elements are listed in increasing order with respect to $\preccurlyeq$. Finally, the operation $\oplus$ is defined by the equations

- $\underline{\infty} \oplus x = x \oplus \underline{\infty} = \underline{\infty}$,
- $(x, y, z) \oplus (x', y', z') = c([(x +^1 x'), (y +^2 y'), (z +^3 z')])$.

For example,

$$(0,2,0) \oplus (0,1,1) = c\big[(0,2,1)\big] = \underline{\infty}$$
$$(0,1,0) \oplus (0,0,1) = c\big[(0,1,1)\big] = (0,1,1).$$

Notice that a Łukasiewicz submonoid is not characterized by its type. For example, assume that the basis of $\boldsymbol{A}^*$ is $[\![0.5, 0.6, 0.69]\!]$. Then the type of $\boldsymbol{A}^*$ is equal with the type of $\boldsymbol{A}$. Nevertheless, $\boldsymbol{A}^*$ is not isomorphic with $\boldsymbol{A}$ since while $\boldsymbol{A}^*$ contains 11 elements $\boldsymbol{A}$ contains 9 elements.

7.3. Product machines

The following proposition holds true.

Proposition 7.3. *Consider a product machine, then order$(x) = \infty$ for every $x \in \boldsymbol{A} - \{0, 1\}$ and, if μ is not crisp, A is infinite. In accordance, the type is a p-sequence as $(\infty, \dots, \infty)$. Also, we have that ordinal$(\boldsymbol{A}) = \omega + 1$.*

Proof. The first part of the proposition is trivial. To prove that $ordinal(\boldsymbol{A}) = \omega + 1$, observe that, since $ordinal(\boldsymbol{A})$ is infinite, this claim is equivalent to the claim that, for every $\underline{x} \in \boldsymbol{A}$, $\{x \in A : x \leq_d \underline{x}\} = \{x \in A : x \geq \underline{x}\}$ is finite. Now, put

$$S_i = \{n \in N : \text{ there are } \underline{n}(1), \dots, n(i-1), n(i+1), \dots, \underline{n}(p) : b_1^{\underline{n}(1)} \cdot \ldots \cdot b_{i-1}^{\underline{n}(i-1)} \cdot b_i^n \cdot b_{i-1}^{\underline{n}(i+1)} \cdot \ldots \cdot b_p^{\underline{n}(p)} \geq \underline{x}\}$$

and assume that S_i is infinite. Then $\underline{x} \neq 1$ and there is an one-to-one function $g : N \to S_i$ such that the sequence $(b_i^{g(n)})_{n \in N}$ has infinite elements. Since $\lim_{n\to\infty} b_i^n = 0$, this entails that there is k such that $b_i^{g(k)} < \underline{x}$. Then $b_1^n \cdot \ldots \cdot b_{i-1}^{\underline{n}(i-1)} \cdot b_i^{g(k)} \cdot b_{i-1}^{\underline{n}(i+1)} \cdot \ldots \cdot b_p^{\underline{n}(p)} \leq b_i^{g(k)} < \underline{x}$ and this contradicts the fact that $g(k) \in S_i$. $\square$

Assume that $dim(\boldsymbol{A}) = p$, then $F^{\underline{o}} = ([0, \infty] \times \ldots \times [0, \infty]) \cup \{\underline{\infty}\}$. Moreover,

$$x +^{\underline{o}} \underline{\infty} = \underline{\infty} +^{\underline{o}} x = \underline{\infty} \text{ and } \underline{\infty} \leq^{\underline{o}} x, \text{ for every } x \in F^{\underline{o}}$$
$$(n_1, \dots, n_p) +^{\underline{o}} (m_1, \dots, m_p) = (n_1 + m_1, \dots, n_p + m_p)$$
$$(n_1, \dots, n_p) \leq^{\underline{o}} (m_1, \dots, m_p) \Leftrightarrow n_1 \leq m_1, \dots, m_p \leq m_p.$$

Also, the kernel $\equiv$ of the canonical epimorphism is defined by the equivalences

$$\big(n(1), \dots, n(p)\big) \equiv \underline{\infty} \Leftrightarrow b_1^{n(1)} \cdot \ldots \cdot b_p^{n(p)} = 0 \Leftrightarrow \text{ there is } i \text{ such that } n(i) = \infty$$
$$\begin{aligned}\big(n(1), \dots, n(p)\big) \equiv \big(m(1), \dots, m(p)\big) &\Leftrightarrow b_1^{n(1)} \cdot \ldots \cdot b_p^{n(p)} = b_1^{m(1)} \cdot \ldots \cdot b_p^{m(p)} \\ &\Leftrightarrow \log\big(b_1^{n(1)} \cdot \ldots \cdot b_p^{n(p)}\big) = \log\big(b_1^{m(1)} \cdot \ldots \cdot b_p^{m(p)}\big) \\ &\Leftrightarrow n(1) \cdot \log(b_1) + \ldots + n(p) \cdot \log(b_p) = m(1) \cdot \log(b_1) + \ldots + m(p) \cdot \log(b_p).\end{aligned}$$

This shows that if $[x] \neq [\underline{\infty}]$, then $[x]$ is the set of points with coordinate in N_0 of a hyperplane like

$$x_1 \cdot \log(b_1) + \ldots + x_p \cdot \log(b_p) = c.$$

Consider the case in which the dimension is 2 and denote by $[\![a, b]\!]$ the basis of $\boldsymbol{A}$. Then, to analyze the equivalence $\equiv$ in $F^{\underline{o}} = ([0, \infty] \times [0, \infty]) \cup \{\underline{\infty}\}$ we observe that,

$$(n, m) \equiv (\underline{n}, \underline{m}) \Leftrightarrow a^n \cdot b^m = a^{\underline{n}} \cdot b^{\underline{m}} \Leftrightarrow n + m \cdot \log_a(b) = \underline{n} + \underline{m} \cdot \log_a(b) \Leftrightarrow n - \underline{n} = (\underline{m} - m) \cdot \log_a(b)$$

where $\log_a(b) > 0$. Now, we have to distinguish two cases.

Case 1. $\log_a(b)$ is not a rational number. Then, equation $n - \underline{n} = (\underline{m} - m) \cdot \log_a(b)$ is possible only in the case $\underline{m} = m$ and therefore $\underline{n} = n$. Then the restriction of $\equiv$ to $[0, \infty) \times [0, \infty)$ is the identity relation while $[\underline{\infty}] = \{(n, \infty) : n \in N_\infty\} \cup \{(\infty, n) : n \in N_\infty\} \cup \{\underline{\infty}\}$. This means that $D = ([0, \infty) \times [0, \infty)) \cup \{\underline{\infty}\}$ and $\oplus$ is defined by the equations

- $\underline{\infty} \oplus x = x \oplus \underline{\infty} = \underline{\infty}$,
- $(n, m) \oplus (\underline{n}, \underline{m}) = (n + \underline{n}, m + \underline{m})$.

Case 2. $\log_a(b) = p/q$. Then,

$$(n, m) \equiv (\underline{n}, \underline{m}) \Leftrightarrow n - \underline{n} = (\underline{m} - m) \cdot p/q \Leftrightarrow q \cdot (n - \underline{n}) = p \cdot (\underline{m} - m).$$

In particular, if $n = k \cdot p + \underline{n}$ and $m = \underline{m} - k \cdot q$, $\underline{m} - m = k \cdot q$, then $q \cdot (n - \underline{n}) = q \cdot k \cdot p = p \cdot k \cdot q = p \cdot (\underline{m} - m)$ and therefore $(n, m) \equiv (\underline{n}, \underline{m})$. This shows that $\equiv$ is different from the identity relation.

In both the cases, since $\log_a(x)$ is an order reversing function,

$$(n, m) \preccurlyeq (\underline{n}, \underline{m}) \Leftrightarrow a^n \cdot b^m \le a^{\underline{n}} \cdot b^{\underline{m}} \Leftrightarrow \log_a\left(a^n \cdot b^m\right) \ge \log_a\left(a^{\underline{n}} \cdot b^{\underline{m}}\right) \Leftrightarrow n + m \cdot \log_a(b) \ge \underline{n} + \underline{m} \cdot \log_a(b)$$
$$\Leftrightarrow n - \underline{n} \ge (\underline{m} - m) \cdot \log_a(b). \qquad \square$$

Observe that, as in the case of Łukasiewicz submonoids, two product submonoid with the same type are not necessarily isomorphic.

We do not go forward in this analysis since the purpose of this section is only to give examples for possible future investigations.

7.4. T-machines

We will examine the more general case of the T-machines. Taking in account of Theorem 2.5 and of the fact that $\boldsymbol{A}$ is finitely generated, the product in $\boldsymbol{A}$ is an ordinal sum of a finite number of triangular norms coinciding either with a Gödel, or with a Łukasiewicz or with a product triangular norm.

Proposition 7.4. *Consider a T-machine and assume that* $\boldsymbol{A}$ *is the sum of continuous triangular norms* $\otimes_1, \dots, \otimes_p$ *in the intervals* $(l_0, l_1), (l_1, l_2), \dots, (l_{p-1}, l_p)$ *where* $0 = l_0 < l_1 < l_2 < \dots < l_p = 1$. *Then, for every* $x \in A$,

$$order(x) = order\big(f_i(x)\big) \text{ where } x \in [l_{i-1}, l_i].$$

In particular, $order(l_i) = order(r_i) = 0$.

Proof. It is sufficient to observe that f_i is an isomorphism. $\square$

There is no difficulty to calculate the dimension, the type and ordinal of A. However, it is necessary to distinguish several cases. For example, consider in every interval (l_{i-1}, l_i) an element b_i and assume that $\boldsymbol{A}$ is the product submonoid generated $b_1, \dots, b_p$. Then, taking in account of the fact that l_{i-1} is not a power of b_i, no element $l_i \in A$ for $i \neq 0$ and $i \neq 1$. Also, is evident that $[\![b_1, \dots, b_p]\!]$ is a basis for $\boldsymbol{A}$ and therefore that $dim(\boldsymbol{A}) = p$. Type of $\boldsymbol{A}$ is the p-ple $(\infty, \dots, \infty)$, its ordinal is $p \cdot \omega + 1$.

8. An observation: fuzzy Turing machines and Church thesis for fuzzy computability

Even though the main purpose of this paper is to give a numerical representation of the fuzzy Turing machines, in this section we will emphasize that Proposition 4.5 enables us to highlight several limits in the computational power of the fuzzy Turing machines. These limits shows that these machines cannot be a candidate for a “Church thesis” in fuzzy set theory. Indeed, as an immediate consequence of this proposition, we have the following theorem.

Theorem 8.1. *Given a fuzzy Turing machine recognizing a fuzzy language e, in Equation (3.1) the value $e(w)$ is attained as a maximum. Moreover, e admits a maximum, i.e. there is an element m such that $s(m) \geq s(x)$ for every $x \in I^+$.*

Proof. Consider Definition 3.2, and observe that both the sets $\{D((S_0/r(1), \ldots, S_n/r(n))) : S_0 = S(w),\ S_n$ is an accepting configuration$\}$ and *values*(e) are subsets of a finitely generated conjunction monoid and therefore that both admit a maximum. □

Now, the fact that a fuzzy subset s which has not a maximum cannot be accepted by a FT-machine strongly contrasts with our intuition. For example, consider the vague predicate 'Big' considered in the heap paradox. If we denote by HEAP the set of all the heaps of sand, it is evident that every time we have a heap x, there is another heap which is strictly bigger than x. Then, no fuzzy subset *big*: HEAP → [0, 1] representing this predicate has a maximum. On the other hand, there are fuzzy subsets able to represent 'Big' which are computable with respect to the classical theory of computability. For example, it is possible to consider the fuzzy subset defined by setting

$$big(x) = \frac{\boldsymbol{g(x)}}{\boldsymbol{g(x)+1}} \tag{8.1}$$

where $g(x)$ is the number of grains in x. It is evident that *big* is a Turing-computable function. Then we have the paradox for which while this fuzzy subset is not recognizable by a FT-machine, it is computable by a classical Turing machine.

Another fact against the adequateness of the fuzzy Turing machines is that the notion of recognizability of the fuzzy subsets with values in [0, 1] depends on the choice of the triangular norm in a strong way. For example, consider a Turing machine $M = (Q, T, I, \Delta, q_0, q_f, \underline{\Delta})$ where $\underline{\Delta} \subseteq \Delta$ is the set of admitted transition rules and assume that the language accepted by this machine is not decidable. Then we can define the fuzzy language *efficiency*: $I^+ \to [0, 1]$ by setting, for every $w \in I$,

$$efficiency(w) = \begin{cases} 0 & \text{if } M \text{ does not converge in } w \\ 2^{-n} & \text{if } n \text{ is the minimum natural number such that } M \text{ converges in} w \text{ in } n \text{ steps.} \end{cases}$$

Then, this fuzzy language is recognizable by a product machine but it is not recognizable by a Zadeh machine or a Łukasiewicz machine. Indeed, let $\mathcal{F}^* = (Q, T, I, \Delta, q_0, q_f, \boldsymbol{A}, \mu)$ be a product machine whose monoid is generated by 1/2 and in which $\mu(\delta) = 1/2$ for every $r \in \underline{\Delta}$ and $\mu(r) = 0$ otherwise. Also, observe that if $S_0 = S(w)$, then $(S_0, \ldots, S_n)$, is an accepting computational path for M if and only if the sequence $(S_0/2^{-1}, \ldots, S_n/2^{-1})$ is an accepting computational path for $\mathcal{F}^*$. In this case $D((S_0/2^{-1}, \ldots, S_n/2^{-1})) = 2^{-n}$. In the case M does not converges in w, then there is no difficulty to find an accepting computational path $(S_0/0, \ldots, S_n/0)$ for $\mathcal{F}^*$. In this case $D((S_0/0, \ldots, S_n/0)) = 0$. Then it is evident that *efficiency* is the fuzzy language accepted by $\mathcal{F}^*$. On the other hand, *values* (*efficiency*) is infinite. Indeed otherwise, if 2^{-m} is the minimum element in this set, given $w \in I^+$,

$$\begin{aligned} M \text{ converges in } w \Leftrightarrow\ & \text{if and only if } M \text{ converges in } w \text{ in } n \text{ steps with } 2^{-n} \geq 2^{-m} \\ \Leftrightarrow\ & \text{if and only if } M \text{ converges in } n \leq m \text{ steps in } w. \end{aligned}$$

This contradicts the fact that the language accepted by M is not decidable. The fact that *values* (*efficiency*) is infinite excludes that the fuzzy subset *efficiency* is recognizable by a Zadeh machine or a Łukasiewicz machine.

Finally, we emphasize that there is no universal fuzzy Turing machine, in general. Namely, in Yongming (2009) one proves that if we refer to the class of bounded lattices and not only to [0, 1], then being finite is a necessary and sufficient condition for the existence of a universal fuzzy Turing machine. This is a disturbing fact in computability theory, obviously.

In my opinion these are evidences that the notion of fuzzy Turing machine is not sufficiently strong to represent computability in fuzzy set theory (see also [8]). In accordance, in [5] and [3] one proposes a different notion of computability in terms of limits and approximation. We recall this notion in the basic case of the interval [0, 1].

Definition 8.2. Denote by $[0,1]_Q$ the set of rational numbers in [0, 1]. Then we say that a fuzzy subset $s : I^+ \to [0,1]$ is *l-semidecidable* provided that there is a computable function $h : I^+ \times N \to [0,1]_Q$ order-preserving with respect to the second variable such that

$$s(x) = \sup\{h(x,n) : n \in N\} = \lim_{n\to\infty} h(x,n). \tag{8.2}$$

We say that a fuzzy subset s is *l-decidable*, provided that it is l-semidecidable and there is an order-reversing computable function $k : I^+ \times N \to [0,1]_Q$ such that

$$s(x) = \inf\{k(x,n) : n \in N\} = \lim_{n\to\infty} k(x,n). \tag{8.3}$$

In particular, if s is l-decidable then $s(x)$ is a computable real number for every $x \in I^+$. This *limit-based* notion of effectiveness does not involve operations to interpret the conjunction and the negation and it is easily extendible to a large class of complete lattices (see [7]). Moreover, it is a proper extension of the classical one and it is a proper extension of the definition given by the fuzzy Turing machines. Observe that the presence of a limit in (8.2) and (8.3) is a crucial fact and it is not surprising. Indeed, in classical logic the effectiveness is connected to recursive arithmetic in account of the discrete structure of the two elements Boolean algebra {0, 1}. So, it is natural to admit that in fuzzy logic the effectiveness has to be connected to recursive analysis in account of the fact that the set [0, 1] of the truth values is a continuum. I emphasize also that this idea of effectiveness is a tool to face a difficulty of fuzzy logic emerged in the paper by B. Scarpellini [17]. In this paper one proves that in Łukasiewicz first order logic the set of tautologies is not semidecidable and therefore that a completeness theorem is impossible (see also [10] and Chapter 11 of [5]). In fact, it is sufficient to refer to a natural definition of fuzzy subset of tautologies and to Definition 8.2.

By concluding, I am convinced that it is possible to propose a sort of '*Church thesis*' for fuzzy logic:

Extended Church thesis The limit-based definition of effectiveness is completely adequate for fuzzy set theory.

In favor of this thesis there is the fact that a *complete* fuzzy subset $k : I^+ \to [0,1]$ exists. This means that k is l-semidecidable and, for every l-semidecidable fuzzy subset $s : I^+ \to [0,1]$ there is a computable one-to-one map h such that $s(x) = k(h(x))$ for every $x \in I^+$.

Once we accept this thesis, a further open question is to search for a definition of fuzzy machine which is in accordance with the limit-based notion of computability. In other words, we have to search for a definition of fuzzy machine plying the same role of the Turing machines in classic mathematics. I am convinced that if it can be done, then we have to imagine fuzzy machines able to perform an endless computation giving as an output an infinite sequence of approximations of the actual belonging degree.

9. Some observations and open questions

All the results in this paper refers to the fuzzy Turing machines. Now, as a matter of fact, these results derive directly from properties of the FT-monoids and from the structure of Formula 3.2. This entails that they hold true also for a rather large class of notions. The fuzzy languages generated by a fuzzy grammar give the main example.

Definition 9.1. Let $\boldsymbol{A}$ be a complete conjunction monoid, then a *fuzzy grammar* is a structure $\boldsymbol{G_A} = (V_n, I, P, s_0, \boldsymbol{A}, \mu)$ such that

- V_n and I are finite disjoint sets of *non-terminal* and *terminal* symbols, respectively,
- $P \subseteq (I \cup V_n)^+ \times (I \cup V_n)^+$ is a finite set of *production rules*,
- $s_0 \in V_n$ is the *start symbol*,
- $\mu : P \to A$ is a fuzzy subset of *production rules*.

If $\boldsymbol{A}$ is total, we say that G is a *total fuzzy grammar*.

Then a fuzzy grammar is a classical grammar $G = (V_n, I, P, s_0)$ together with a fuzzy subset μ of P to evaluate the correctness degree of a production rule. We denote by $\alpha \to \beta$ a production rule and by $\alpha \xrightarrow{\rho} \beta$, where $\rho = \mu(\alpha, \beta)$, a *fuzzy production rule* and we write $x \overset{\rho}{\Rightarrow} y$ if there is a fuzzy production rule $\alpha \overset{\rho}{\rightarrow} \beta$ such that $x = z_1\alpha z_2$ and $y = z_1\beta z_2$. In this case we say that x *directly derives* y. A *derivation chain from* s_0 to w is a sequence

$$\big(s_0 \overset{\rho(\mathbf{0})}{\Longrightarrow} \alpha_1, \alpha_1 \overset{\rho(\mathbf{1})}{\Longrightarrow} \alpha_2, \ldots, \alpha_m \overset{\rho(\boldsymbol{m})}{\Longrightarrow} w\big).$$

Also, the *validity degree* of a derivation chain is defined by setting

$$D\big(s_0 \overset{\rho(\mathbf{0})}{\Longrightarrow} \alpha_1, \alpha_1 \overset{\rho(\mathbf{1})}{\Longrightarrow} \alpha_2, \ldots, \alpha_m \overset{\rho(\boldsymbol{m})}{\Longrightarrow} w\big) = \rho(0) \cdot \ldots \cdot \rho(m).$$

Definition 9.2. The *fuzzy language generated by the fuzzy grammar* G is the fuzzy subset $e : I^+ \rightarrow A$ defined by setting, for every $w \in I^+$,

$$e(w) = \sup\big\{D(Z) : Z \text{ is a derivation chain from } s_0 \text{ to } w\big\}. \tag{9.1}$$

Now, Equation (9.1) has the same structure of Equation (3.1). Both define a fuzzy subset by the least upper bound of a finite product of values in a conjunction monoid. This enables us to extend all the results in this paper to the fuzzy grammars. Further examples are furnished by fuzzy control and fuzzy logic programming (see [19] and [6]).

Another interesting question is the possibility of extending the results in this paper to fuzzy machines whose evaluation structure involves operators different from the product. This is done, for example, by E.S. Santos in [16] where one stars from an ordered semiring $(A, \otimes, \oplus, \leq)$ and one defines the fuzzy language accepted by a fuzzy machine by a formula involving both the operators $\otimes$ and an infinitary extension of $\oplus$. A strong difference is that we cannot extend Proposition 4.5 to these structures as the following counterexample shows.

Proposition 9.3. *Assume that* $\boldsymbol{A} = ([0, 1], \otimes, \oplus, \leq, 0, 1)$ *is the semiring in which* $\otimes$ *is the usual product and* $\oplus$ *the associated co-norm defined by setting* $x \oplus y = x + y - x \cdot y$. *Then there are finitely generated substructures of* $\boldsymbol{A}$ *containing a nonempty subset with no maximum.*

Proof. Define $f : [0, 1] \rightarrow [0, 1]$ by setting $f(x) = 1 - x$, then f is an isomorphism among the reducts $([0, 1], \oplus, 0, 1)$ and $([0, 1], \otimes, \leq, 1, 0)$. Define the multiple of an element with respect to $\oplus$ as usual, i.e. $1x = x$ and $mx = ((m - 1)x) \oplus x$. Then since $f(x \oplus y) = f(x) \otimes f(y)$, we have that $f(mx) = f(x)^m$ and therefore $mx = 1 - (1 - x)^m$. By setting $x = 1/2$, we obtain that $m(1/2) = 1 - (1/2)^m$. This means that $\{1 - (1/2)^m : m \in N\}$ is a subset of the semiring generated by $1/2$ in which there is no maximum. □

Then, probably the fuzzy machines defined on a semiring are able to accept a class of fuzzy languages larger than the one of the fuzzy language accepted by the fuzzy machines defined on a conjunction monoid. Nevertheless, I still believe that this class is not adequate for a Church thesis in the fuzzy framework. This since I doubt that all the computable functions from I^+ to $[0, 1]$ can be obtained from the two operations $\otimes$ and $\oplus$. Should be interesting to prove or to disprove this conjecture.

References

[1] B.R.C. Bedregal, S. Figueira, Classical computability and fuzzy Turing machines, Fuzzy Sets Syst. 159 (2008) 1072–1083.
[2] R. Belohlavek, Pavelka-style fuzzy logic in retrospect and prospect, Fuzzy Sets Syst. 281 (2015) 61–72.
[3] L. Biacino, G. Gerla, Fuzzy logic, continuity and effectiveness, Arch. Math. Log. 41 (2002) 643–667.
[4] K. Evans, M. Konikoff, R. Mathis, J. Madden, G. Whipple, Totally ordered commutative monoids, Semigroup Forum 62 (2) (2001) 249–278.
[5] G. Gerla, Fuzzy Logic: Mathematical Tools for Approximate Reasoning, Kluwer Academic Press, 2000.
[6] G. Gerla, Fuzzy logic programming and fuzzy control, Stud. Log. 79 (2005) 231–254.
[7] G. Gerla, Effectiveness and multi-valued logics, J. Symb. Log. 71 (2006) 137–162.
[8] G. Gerla, Comments on some theories of fuzzy computation, Int. J. Gen. Syst. 45 (4) (2016) 372–392.
[9] S. Gottwald, A Treatise on Many-Valued Logics, Research Studies Press LTD, Baldock, Hetfordshire, England, 2001.
[10] P. Hájek, Metamathematics of Fuzzy Logic, Kluwer Academic Publishers, Dordrecht, 1998.
[11] E.P. Klement, R. Mesiar, E. Pap, Triangular Norms, Trends in Logic, Kluwer Editors, 2000.
[12] G.J. Klir, T. Folger, Fuzzy Sets, Uncertainty, and Information, Prentice Hall, Englewood Cliffs, NJ, USA, 1987.
[13] G.J. Klir, B. Yuan, Fuzzy Sets and Fuzzy Logic: Theory and Applications, Prentice Hall, 1995.
[14] V. Novák, I. Perfilieva, J. Močkoř, Mathematical Principles of Fuzzy Logic, Kluwer Academic Publishers, London, 1999.

[15] H. Rogers, Theory of Recursive Functions and Effective Computability, McGraw-Hill, New York, 1967.
[16] E.S. Santos, Fuzzy algorithms, Inf. Control 17 (1970) 326–339.
[17] B. Scarpellini, Die Nichaxiomatisierbarkeit des unendlichwertigen Prädikatenkalküls von Łukasiewicz, J. Symb. Log. 27 (1962) 159–170.
[18] A. Syropoulos, Theory of Fuzzy Computation, IFSR International Series on Systems Science and Engineering, Springer, New York, 2014.
[19] P. Vojtáš, Fuzzy logic programming, Fuzzy Sets Syst. 124 (2001) 361–370.
[20] J. Wiedermann, Fuzzy Turing machines revised, Inf. Comput. 21 (2002) 1–13.
[21] J. Wiedermann, Characterizing the super-Turing computing power and efficiency of classical fuzzy Turing machines, Theor. Comput. Sci. 317 (2004) 61–69.
[22] L. Yongming, Lattice-valued fuzzy Turing machines: computing power, universality and efficiency, Fuzzy Sets Syst. 160 (2009) 3453–3474.