Mathware & Soft Computing 6 (1999) 163-172

Towards a Fuzzy Computability?

C. Moraga
Dept. Computer Science, Comp. Engineering and Comp. Education
University of Dortmund
44221 Dortmund Germany
moraga@cs.uni-dortmund.de

Abstract

The subject of the present paper is the study of fuzzy computability based on fuzzy Turing machines. Two different models of fuzzy Turing machines will be discussed. It is shown that most work on fuzzy mathematics may be conducted within the frame of classical computability and the rest falls within the area of computability of the reals.

1 Introduction

In its three and a half decades of existence since the seminal paper of L.A. Zadeh [Zad 65], fuzzy logic has absolved the stages of wide theoretical research, hardware-supported fuzzy systems and industrial applications, (the later, mainly in the area of fuzzy control).

Activities in the area of hardware-supported fuzzy systems may be traced back to work initiated in the middle eighties by M. Togai and H. Watanabe [ToW 86] as well as T. Yamakawa [Yam 88]. The expectations to see the development of fuzzy computers, were however not fulfilled. Possible exceptions may be some university projects, like the ORBE-Project in Spain [Mor 93], [Rui 94], [Sal 96]. Most hardware efforts conveyed very fast to fuzzy control (see e.g. [BBR 93], [Bat 96]). A significant amount of industrial available "fuzzy processors" of the present days are mainly programmable fuzzy controllers. There is however still lot of work being done in this area, as may be observed in presentations and special sessions in most large conferences of the fuzzy community (see e.g. the Proceedings of IPMU'96/98, FUZZ-IEEE'97/99, IFSA'97/99, EUFIT'98/99, Iizuka'98) and has been enhanced by the appearance of the first books on the subject (see e.g. [KaL 98]).

Fuzzy algebra Fuzzy dynamics and chaos Fuzzy mathematical programming Fuzzy measures and integrals Fuzzy multivariate analysis Fuzzy numbers Fuzzy optimization Fuzzy relations Fuzzy sets theory Fuzzy statistics Fuzzy topology and analysis

Table 1: Selection of session-topics of the IFSA'97 Conference

Fuzzy control has become the possibly best known "result" of fuzzy logic due to its industrial acceptance. Even though fuzzy control is without any doubt an important area, it is not the only one that has been given support by the scientific community. See for instance Table 1, which shows a summary of some of the session-subjects covered during the recent IFSA Conference 1997 in Prague. There are impressive "mathematically oriented" developments beyond fuzzy control. This motivates the initial question of this paper: Do we need a (new) concept for fuzzy computability?

The rest of the paper is organized as follows. In the next section the basics of computability will be shortly presented. Fuzzy Turing Machines will be the subject of the third section. A section of conclusions will close the paper.

2 Crisp computability

Basic to the study of computability is the concept of *algorithm* (traced back to work done in the IX century by the Persian mathematician Al Chowarizmi, on procedures for the formal solution of systems of equations. This might well have been the origin of the word *algorithm*. See e.g. [Rec 91]). Nowadays, unseparable from the algorithm concept is that of the Turing Machine [Tur 36], which is acknowledged as a computing model with a power equivalent to that of algorithms. That is, if a function may be evaluated in terms of whatever kind of algorithm then there exists a Turing machine that computes the same function. This is known as Church's Thesis [Chu 36] and the former function will be said to be "Turing computable".

From the many ways of defining a Turing machine (see e.g. [LeP 81], [Rec 91], [Weg 93], [URL 01]), the following is probably one of the simplest.

Definition 1: Turing machine

$$\mathrm{M} = (\mathrm{S}, \mathrm{Q}, \mathrm{q}_0, \mathrm{E}, \delta\ ) \quad (1)$$

where S represents a finite non-empty set of input symbols,

Q denotes a finite non-empty set of states, with $\mathrm{S} \cap \mathrm{Q} = \emptyset$

$q_0 \in Q$ is the symbol to designate the initial state of the machine

$E \subset Q$ is a finite non-empty set of final states

and

$$\delta : Q \times S \rightarrow Q \times (S \cup \{\text{right, left }\}) \quad (2a)$$

or

$$\delta : Q \times S \rightarrow P\ [\ Q \times (S \cup \{\text{right, left}\}\ ] \quad (2b)$$

denotes the transition function. Eq. 2a represents the case of a deterministic Turing machine meanwhile eq. 2b, that of a non-deterministic one (with P[X]) representing the power set of (some given set) X). Furthermore, an especial symbol $\# \in S$ will be agreed upon to represent an "empty information" at the input.

A Turing machine may be imagined to have an infinite input tape that may contain a word $w \in (S \backslash \#)^*$ and otherwise escorting strings of # symbols. A read-write head interacts with the tape and a central unit controls the work of the machine (according to δ). To begin with the work, the machine is in state q_0 and the head is placed on the last # before the beginning of a word w (assuming, without loss of generality, that the word will be read from left to right). The head moves one place to the right, reads the first (non-#) symbol and evaluates δ. As a consequence, the machine may keep or change its state, may keep or change the symbol under the head or else may either move the head one place to the right or to the left (and compute again δ). If the Turing machines reaches a state q_x and δ (q_x, s) is not specified (for any $s \in S$) then it stops. If the state of the machine when it stops belongs to E then it returns implicitly a "yes". It returns a "no" if the state of the machine does not belong to E. If "no", the machine rejects the word. If "yes", the word has been accepted and the processed result is written on the tape. Finally it is possible that a Turing machine does not stop at all on a given input word w.

In order to formally analyze the work of a Turing machine, the concepts of configuration and computation will be introduced.

A string uqav where $a \in S$ represents the symbol under the head, u,v $\in S^*$ represent the prefix and suffix of a respectively and q denotes the present state of the machine, is called a configuration. (See figure 1).

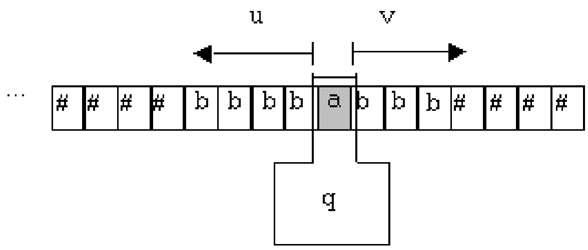

Fig. 1: Turing machine with the configuration C = uqav

If a Turing machine is in a configuration $C_i = u_i\ q_i\ a_i\ v_i$ and after evaluating $\delta(a_i, q_i)$ changes to the configuration $C_j = u_j\ q_j\ a_j\ v_j$, then the configuration C_j is direct reachable from C_i. This will be written $C_i \rightarrow C_j$. If there exists a sequence of pairwise direct reachable configurations $C_i \rightarrow C_{i+1} \rightarrow C_{i+2} ... \rightarrow C_j$, then the configuration C_j is (indirect) reachable from C_i and this will be expressed as $C_i \rightarrow^* C_j$, where $\rightarrow^*$ denotes the transitive closure of $\rightarrow$. Finally if C_0 denotes an initial configuration and C_n a final one (in the sense that the Turing machine stops at this configuration and $q_n \in E$), then $C_0 \rightarrow^* C_n$ is called a computation of the machine.

3 Fuzzy computability

Basic for the study of fuzzy computability is the concept of fuzzy algorithm that was introduced in [Zad 68], but not further developed. As in the case of crisp computability, alternative to fuzzy algorithms, fuzzy Turing machines may be considered. Two models of fuzzy Turing machines will be discussed below. In the first model, all sets of the definition expressed in eq. (1) will be changed into corresponding fuzzy sets. The transition function δ will then become a function over fuzzy sets. In the second model, the sets will not be changed, but a fuzzifying function will be associated to δ.

The following notation will be used in the rest of the paper. Fuzzy sets will be expressed and named by their membership function. Fuzzy sets will be defined in given universes of discourse and the range of their membership functions will be [0,1] or $[0,1]_C$, where $[0,1]_C$ denotes the subset of computable real numbers in the interval [0,1]. The names of fuzzy sets will be written in *italics* meanwhile for crisp sets regular fonts will be used. Given a fuzzy set B defined in the universe U, the value of the membership function at a given place $p \in U$ is given by $B(p)$. Furthermore an element of a fuzzy set B may be interpreted as pair (identifier, membership), that formally is an element of the Cartesian product $\text{supp}(B) \times [0,1]$. (Where $\text{supp}(B) = \{ p \in U — B(p) > 0 \}$). Accordingly, $B \subseteq \text{supp}(B) \times [0,1]$.

1st model

Definition 2: A fuzzy Turing machine $M1$ is a 5-tuple (S, Q, q_0, E, δ), where

S is a fuzzy set over a universe U_S of symbols. S: $U_S \rightarrow [0, 1]_C$. $\# \in S$ with $S(\#) = 1$. The support of S is finite

Q is a fuzzy set over a universe U_Q of states. Q: $U_Q \rightarrow [0,1]_C$. The support of Q is finite. $S \cap Q = \emptyset$.

The initial state q_0 has the following properties: $q_0 \in Q$ with $Q(q_0)=1$

E is a crisp finite non-empty subset of Q

and

$$\delta : Q \times S \rightarrow \text{supp}(Q) \times (\text{supp}(S) \cup \{\text{left, right}\}) \quad (3a)$$

Notice that at the right hand side of the definition of δ, only the support of Q and S are required. As soon as the identifiers for the (possibly) new state and symbol are respectively obtained, their corresponding membership degrees may be directly calculated. In this way, δ is a function with a (computable) fuzzy domain but a crisp range. Moreover the fuzzy sets of the domain may be represented as the Cartesian product of their support and $[0,1]_C$ thus leading to

$$\delta : (\mathrm{supp}(Q) \times [0,1]_C) \times (\mathrm{supp}(S) \times [0,1]_C) \rightarrow \mathrm{supp}(Q) \times (\mathrm{supp}(S) \cup \{\mathrm{left, right}\}) \quad (3b)$$

The non-deterministic version is given by

$$\delta \subset (Q \times S) \times (\mathrm{supp}(Q) \times (\mathrm{supp}(S) \cup \{\mathrm{left, right}\})) \quad (4a)$$

or

$$\delta \subset (\mathrm{supp}(Q) \times [0,1]_C) \times (\mathrm{supp}(S) \times [0,1]_C) \times (\mathrm{supp}(Q) \times (\mathrm{supp}(S) \cup \{left, right\})) \quad (4b)$$

One transition of the fuzzy Turing machine $M1$ is illustrated in figure 2.

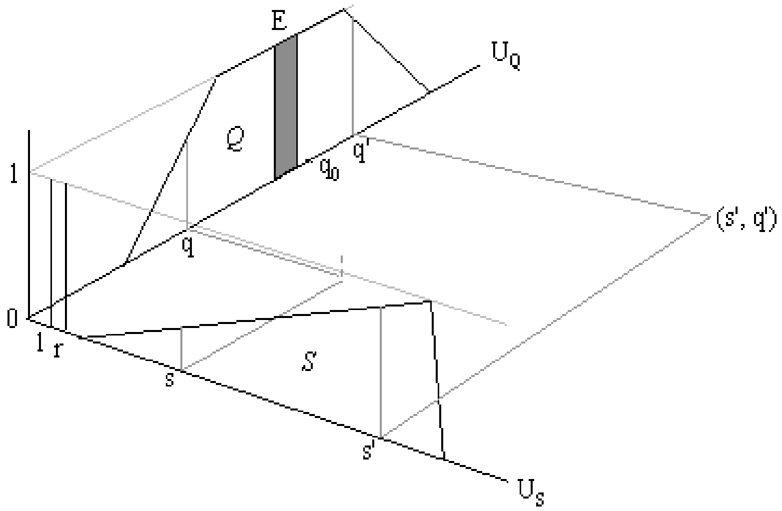

Fig. 2: A possible transition of the fuzzy Turing machine $M1$

Finally, the result of a computation of the fuzzy Turing machine will be defined as follows:

$$\rho = \left(w, \prod_{i=1}^{n} S(w_{\mathrm{i}}) \right), \quad (5)$$

where $w \in (S\backslash\#)^*$ denotes the word on the tape, $\prod$ represents the transitive closure of a predefined t-norm and $\mathrm{w_i}$ stands for the i-th symbol of the result-word. Equation (5) associates to the meaning of the word w a *degree of certainty* computed as a function of the membership degrees of its symbols.

Algorithm A1 describes a computation with the fuzzy Turing machine $M1$. Then there exists a (crisp) Turing machine M1 that computes Algorithm A1. It follows that M1 is equivalent to $M1$.

In conclusion: for any fuzzy Turing machine constructed after Model 1, there exists an equivalent crisp Turing machine. Furthermore, problems solved with the fuzzy Turing machine may then be studied within the frame of classical computability.

Algorithm A1;
q = q_0
while q $\notin$ E
  remember q; read s; evaluate Q(q) and S(s);
  compute (q', s') = δ ((q, Q(q)),(s, S(s));
  if s'({ left, right} then move accordingly else
  s= s';
  q=q';
compute the degree of certainty of the result
end.

2nd Model

W-functions were introduced by E. Santos in the late seventies [San 77] and used to study W-Turing machines. For the present model, a slightly different version of W-Functions will be used.

Definition 3: W-functions

Let f: U $\rightarrow$ V, where U and V are non-empty sets. The function f is not required to be total. The W-function f_W associated to f is normally a partial function given by: f_W: (U $\times$ V) $\rightarrow$ W, where W = $[0,1]_C$ and f_W is defined at all pairs (u,v) where f(u) is defined and f(u)=v holds. f_W(u,v) assigns a degree of certainty to the computation of f(u)=v. The following reference structure will be used: $([0,1]_C, \tau, \tau^*)$, where τ is a t-norm and τ^* is a t-conorm, dual with respect to τ. Even though t-norms may be traced back to work done by Karl Menger in the early forties [Men 42] they were discovered by people interested in fuzzy logic after the publication of a book by B. Schweizer and A. Sklar [ScS 83].

Definition 4: Fuzzy Turing machine (in analogy to [San 77])

$M2$ = (S, Q, q_0, E, δ, δ_W) (6)

where S represents a finite non-empty set of input symbols,

Q denotes a finite non-empty set of states, with S $\cap$ Q = $\emptyset$

$q_0 \in$ Q is the symbol to designate the initial state of the machine

E$\subset$Q represents a finite non-empty set of final states

$\delta \subset$ (Q $\times$ S) $\times$ (Q $\times$ (S $\cup$ { right, left })) (7)

and δ_W : (Q $\times$ S) $\times$ (Q $\times$ (S $\cup$ { right, left }) $\rightarrow [0,1]_C$ (8)

It becomes apparent δ_W assigns a degree of certainty to every transition of the machine. Moreover let the concepts of configuration and computation be used here as in the crisp Turing machine.

Definition 5: Degree of reachability

Let $C_i \rightarrow C_{i+1}$. Then $\eta_W(C_i, C_{i+1})$ denotes the degree of reachability of C_{i+1} from C_i.

$\forall$u, v $\in$ S*: a, a' $\in$ S; q, q'$\in$ Q

$$\eta_W(C_i, C_{i+1}) = \begin{cases} \delta_W(q, a, q', a') & \text{if} \quad C_i = uqav \quad C_{i+1} = uq'a'v \\ \delta_W(q, a, q', right) & \text{if} \quad C_i = uqav \quad C_{i+1} = uaq'v \\ & \text{or} \quad C_i = uqa \quad C_{i+1} = uaq'\# \\ \delta_W(q, a, q', left) & \text{if} \quad C_i = uaqv \quad C_{i+1} = uq'av \\ & \text{or} \quad C_i = qav \quad C_{i+1} = q'\#av \\ 0 & \text{if} \quad C_i \not\to C_{i+1} \end{cases} \tag{9}$$

Definition 6: Degree of certainty.

Let Γ denote the degree of certainty of the computation done by the fuzzy Turing machine $M2$. If $C_0 \to^* C_n$, where C_0 and C_n represent an initial and end configuration, respectively, then the degree of certainty of the computation is evaluated as follows:

$$\Gamma(C_0,C_n) = \tau\big[\eta_W(C_0,C_1), \tau[\eta_W(C_1,C_2), ...\tau[\eta_W(C_{n-2},C_{n-1}), \eta_W(C_{n-1},C_n)]...]\big] \tag{10}$$

In the case of a non-deterministic fuzzy Turing machine, there may exist different sequences of pairwise directly reachable configurations leading from C_0 to a given C_n and these different sequences may also have different degrees of certainty. Let $\mathbf{G}_{0,n}$ denote the set of degrees of certainty of the (non-deterministic) computation $C_0 \to^* C_n$. Then:

$$\Gamma(C_0, C_n) = \sum_{\gamma \in G_{0,n}} {}^{*}\gamma \tag{11}$$

where $\sum^*$ denotes the transitive closure of τ^*.

Procedure A2 gives a precise description of the behavior of the fuzzy Turing machine $M2$ when doing a computation. Since all elementary steps in A2 are effectively computable, then A2 is indeed an algorithm and there exists a crisp Turing machine M2 that processes Algorithm A2.

It follows then that there exists a crisp Turing machine M2 which is equivalent to the fuzzy Turing machine $M2$. Since this equivalence has been shown for an arbitrary fuzzy Turing machine, it holds for all fuzzy Turing machines specified by equation (4).

Moreover it may be inferred, that results obtained with the fuzzy Turing machine $M2$ may be further analyzed within the classical computability theory.

```
Procedure A2;
G= ∅;
// simulation of non-determinism under fairness //
for j=1 to 100...0 do
begin
  i=0; Γ0 = 1; q=q0;
  while q∉E
  begin
      (q', s') = δ(q, s);
      Γi+1 = τ[Γi, δW(q, s, q', s')];
      i=i+1; q=q'; s=s'
  end;
  // the machine stops at Cn //
  read result-word and Γi
end;
if Γi ∉G0,n then
  begin
      G0,n =G0,n ∪ {Γi};
      Compute Γ(C0,Cn) with eq. (11)
  end
end Procedure A2.
```

4 Conclusions

Let ***M*** denote the set of fuzzy Turing machines after model 1 or model 2 and let **M** denote the set of classical Turing machines (see Definition 1). Then ***M*** and **M** are equivalent.

The subset $[0,1]_C$ is dense enough to support an adequate representation of real-world problems, based on fuzzy sets. (See e.g. [Wei 87] for numbers in $[0,1]$ $[0,1]_C$).

Let ***M'*** represent the set of fuzzy Turing machines defined after model 1 or model 2 however using the full interval [0,1] as range for the membership function of fuzzy sets or for the fuzzifying transition function δ_W. It follows that ***M***⊂***M'***. The world ***M'*** \ ***M*** is open for further research. The computability aspects here however refer to computability of the reals (Type 2 Computability) [Grz 55], [Wei 87] and do not imply a fuzzy computability. Related aspects have been studied mostly under the keyword W-computability, as in [Cla 83], [ViC 84], [ClD 87], [Ger 89] and [MPC 93].

5 References

[Bat 96] Baturone I.: *Implementación VLSI de Controladores Difusos.* Dissertation, Dept. Electronics and Electromagnetism, University of Sevilla, 1996

[BBR 93] Bugarín A., Barro S., Ruiz R.: Soluciones Hardware en Control Borroso. In *Estudios de Lógica Borrosa y sus Aplicaciones* (A. Sobrino y S. Barro, Eds.), 295-325. Imprenta Universitaria, Universidad de Santiago de Compostela, ISBN 84-8121-018-8, 1993

[Chu 36] Church A.: An unsolvable problem of Elementary Number Theory. *American Journal of Mathematics*, 58, 345-363, 1936

[Cla 83] Clares B.: Una Introducción a la W-Calculabilidad: Operaciones básicas. *Stochastica* **VII**, (2), 111-135, 1983

[ClD87] Clares B., Delgado M.: Introduction to the concept of Recursiveness of Fuzzy Functions. *Fuzzy Sets and Systems* **21**, 301-310, 1987

[Ger89] Gerla G.: Turing L-Machines and Recursive Computability for L-Maps. *Studia Logica* **XLVIII**, (2), 179-192, 1989

[Grz 55] Grzegorczyk A.: Computable functionals. *Fundamenta Mathematica* **42**, 168-202, 1955

[KaL 98] Kandel A., Langholz G. (Eds.): *Fuzzy Hardware. Architectures and Applications.* Kluwer Academic Publishers, Boston, 1998

[LeP 81] Lewis H.R., Papadimitriou Ch.H.: *Elements of the Theory of Computation.* Prentice Hall, Englewood Cliffs, 1981

[Men 42] Menger K.: Statistical Metrics. *Proceedings National Academy of Sciences USA* **28**, 535-537, 1942

[Mor 93] Moraga C.: El Proyecto ORBE. In *Estudios de Lógica Borrosa y sus Aplicaciones* (A. Sobrino y S. Barro, Eds.), 327-332. Imprenta Universitaria, Universidad de Santiago de Compostela, ISBN 84-8121-018-8, 1993

[MPC93] Morales R., Pérez de la Cruz J.L., Clares B., Conejo R.: Estudio de una clase de W-funciones calculables. *Actas 3r. Congreso Español sobre Tecnologías y Lógica Fuzzy*, 153-160, Santiago de Compostela, 1993

[Rec 91] Rechenberg P.: *Was ist Informatik?* Carl Hanser Verlag, Wien, 1991

[Rui 94] Ruiz A.: *Hardware básico de motores de inferencia borrosa utilizando trapecios codificados.* Dissertation, Faculty of Computer Science, Technical University of Madrid, 1994

[Sal 96] de Salvador L.: *Model of a High Performance Fuzzy Microprocessor and its Design.* Dissertation, Dept. of Photonic Technology, Faculty of Computer Science, Technical University of Madrid, 1996

[San77] Santos E.S.: Fuzzy and Probabilistic Programs. In *Fuzzy Automata and Decision Process*, (M.M. Gupta, G.N. Saridis, B.R. Gaines, Eds.), 133-148. North-Holland, Amsterdam, 1977

[ScS 83] Schweizer B., Sklar A.: *Probabilistic Metric Spaces.* North Holland, Amsterdam, 1983

[ToW 86] Togai M., Watanabe H.: Expert System on a Chip: An Engine for real-time Approximate Reasoning. *IEEE Expert*, 55-62, 1986

[Tur 36] Turing A.: On Computable Numbers, with an application to the Entscheidungsproblem. *Proc. Lond. Math. Soc.* **42** (2), 230-265, 1936

[URL 01] URL: http://www.turing.org.uk/turing/biblio.html

[ViC 84] Vila A., Clares B.: On the W-Computability of Fuzzy Functions. *Proceedings Fall International Seminar on Applied Logic, FISAL,* 41-53, Portocolom, Mallorca, Editorial Universitat de les Illes Balears, ISBN 84-600-4583-8, 1984

[Weg 93] Wegener I.: *Theoretische Informatik – eine algorithmenorientierte Einführung.* Teubner, Stuttgart, 1993

[Wei 87] Weihrauch K.: *Computability.* Springer, Heidelberg, 1987

[Yam 88] Yamakawa T.: High-speed Fuzzy Controller Hardware System: The mega-FIPS machine. *Information Sciences*, (45), 113-128, 1988

[Zad 65] Zadeh L.A.: Fuzzy Sets. *Information and Control* **8**, 338-353, 1965

[Zad 68] Zadeh L.A.: Fuzzy Algorithms. *Information and Control* **12**, 94-102, 1968