Iranian Journal of Fuzzy Systems

Volume 16, Number 1, (2019), pp. 49-60

University of Sistan and Baluchestan

Some improvements in fuzzy turing machines

H. Farahani[1,2]

[1] *Department of Computer Science, Shahid Beheshti University, G.C, Tehran, Iran*

h_farahani@sbu.ac.ir

Abstract

In this paper, we improve some previous definitions of fuzzy-type Turing machines to obtain degrees of accepting and rejecting in a *computational* manner. We apply a BFS-based search method and some level's upper bounds to propose a computational process in calculating degrees of accepting and rejecting. Next, we introduce the class of Extended Fuzzy Turing Machines equipped with *indeterminacy* states. These states are used to characterize the loops of classical Turing machines in a *mathematical* sense. In the sequel, as well as the notions of acceptable and decidable languages, we define the new notion of indeterminable language. An indeterminable language corresponds to non-halting runs of a machine. Afterwards, we show that there is not any universal extended machine; which concludes that these machines cannot solve the halting problem. Also, we show that our extended machines and classical Turing machines have the same computational power. Then, we define the new notion of semi-universality and prove that there exists a semi-universal extended machine. This machine can indeterminate the complement of classical halting problem. Moreover, to each r.e or co-r.e language, we correspond a language that is related to some extended fuzzy Turing machines.

Keywords: Theory of computation, Extended fuzzy Turing machine, Indeterminacy state, Halting Problem.

1 Introduction

In [14], Lotfi Zadeh defines the notion of fuzzy algorithm. His definition is based on a fuzzification of Turing machines. However, that work was not deep enough in the recursion theoretical aspects of the mentioned model. That work is followed by the same setting in [7]. The equivalency of previous fuzzy models is shown in [9, 10]. Afterwards, the research in this field is revisited in [2, 4, 6]. In [2], Biacino and Gerla generalize the definition of recursive enumerability introduced in [6]. Next, Wiedermann proposes a formal fuzzy computing model based on Turing machine model [12, 13]. He claims that it is possible to accept r.e. sets and co-r.e. sets by fuzzy Turing machines and so these machines can solve the halting problem. So he claims that these fuzzy Turing machines have more computational power than the classical Turing machines. In [1], Bedregal and Figueira analyse Wiedermann's statement about the computational power of fuzzy Turing machines and show that Wiedermann's statement is not completely correct. They give a characterization of the class of recursively enumerable (r.e) sets in terms of associated fuzzy languages recognized by fuzzy Turing machines. They also show that there is no universal fuzzy Turing machine which can simulate each machine in the class of all fuzzy Turing machines. In [8], Moniri defines the class of Generalized Fuzzy Turing Machines. His machines are equipped with both accepting and rejecting states. He studies some basic computability aspects of his machines and proves that a fuzzy language L is decidable if and only if L and L^c are acceptable. Recently, a meta-type fuzzy Turing machine is defined in which the transitions can take dynamic degrees corresponding with the input [3]. The advantage of this machine over other fuzzy-type Turing machines is that, by applying dynamic degrees of transitions a larger class of fuzzy languages is recognized. Also a fuzzy theory of computational complexity is established.

In Section 3, we make some essential modifications in previous definitions of fuzzy Turing machines in [1, 8, 12]. In these works, degree of accepting or rejecting is not *computationally* well defined and there are some cases which these degrees can not be calculated in a computational process. We modify these notions by (1) applying a BFS-based search

Corresponding Author: H. Farahani
Received: July 2017; Revised: October 2017; Accepted: April 2018.

in the computational tree of a given fuzzy Turing machine on an input, (2) obtaining some upper bounds on the number of levels in our BFS-based search. If we reach an accepting or a rejecting configuration in a level of computational tree, then the upper bounds indicate the number of next levels that are needed to be traversed (at most) to determine the existence of *another* accepting or rejecting configuration.

In Section 4, we establish a new class of fuzzy Turing machines that we call Extended Fuzzy Turing Machines. These machines are equipped with *indeterminacy* states as a new type of states. There are two-sided *silent* transitions between indeterminacy states and all other *non-accept* and *non-reject* states with degree 1 and transitions from accept and reject states to indeterminacy states with degree 0. Indeterminacy states are applied to catch loops of machines on their inputs. The loop on an input is identified when input's indeterminacy degree equals 0. Next, we show that even for extended machines which may loop on some inputs, there exists a function $r(n)$ that we can compute degree of accepting or rejecting of a given input (if exists) in time $2^{O(r(n))}$. We define the new notion of *indeterminable* language. An indeterminable language characterize elements which are neither accepted nor rejected by an extended machine. Intuitively, an indeterminable language contains words which a machine loops on them. Afterwards, we show that there is not a universal extended machine. It makes the extended Turing machines unable to solve the halting problem. Then, we show that our extended machines and classical Turing machines have the same computational power. We define a new notion of semi-universality and show that there is a semi-universal extended machine. This semi-universal machine can indeterminate the complement of classical halting problem. Finally, to each r.e or co-r.e language, a language is corresponded that is related to some appropriate extended fuzzy Turing machines.

2 Preliminaries

In this section, we review some preliminaries from the literature.

Definition 2.1. [15] *A fuzzy subset A of a set X is a function $\mu_A : X \to [0,1]$, where for each $x \in X$, $\mu_A(x)$ represents the grade of membership of the element $x \in X$ to A.*

Definition 2.2. [5] *A t-norm is a binary operation $*$ on $[0,1]$ satisfying commutativity, associativity, non-decreasing in both arguments, with the properties $0 * x = 0, 1 * x = x$, for all x.*

Definition 2.3. [5] *If $*$ is a t-norm, then its dual t-conorm $*'$ satisfies the following condition:*

$$*'(x,y) = 1 - *(1-x, 1-y)$$

Definition 2.4. [8] *Let L_1 and L_2 be two fuzzy languages. Also, let $*$ be a t-norm and $*^{'}$ be its dual t-conorm. The languages $L_1 * L_2$ and $L_1 *^{'} L_2$ are defined as follows:*

$$(L_1 * L_2)(x) = L_1(x) * L_2(x) \qquad (L_1 *^{'} L_2)(x) = L_1(x) *^{'} L_2(x).$$

Definition 2.5. [1] *A fuzzy Turing machine (FTM) is a triple $\mathcal{F} = (\mathcal{T}, *, \mu)$, where $\mathcal{T} = (Q, \Sigma, \Gamma, \Delta, q_s, F)$ is a non-deterministic Turing machine, $*$ is a t-norm and $\mu : Q \times \Gamma \times Q \times \Gamma \times \{R, L\} \to [0,1]$ is a function. In Turing machine $\mathcal{T}$, Q is a set of states, Σ is a set of input symbols, Γ is a set of tape symbols, transition relation Δ is a subset of $Q \times \Gamma \times Q \times \Gamma \times \{R, L\}$, $q_s \in Q$ is the starting state and $F \subseteq Q$ is the set of final states.*

Definition 2.6. [8] *A Generalized fuzzy Turing machine (GFTM) is a tuple $\mathcal{F} = (\mathcal{T}, *, *^{'}, \eta)$, where $\mathcal{T} = (Q, \Sigma, \Gamma, \Delta, q_s, F)$ is a non-deterministic Turing machine such that Q is a set of states, Σ is a set of input symbols, Γ is a set of tape symbols, Δ is a **fuzzy subset** of $Q \times \Gamma \times Q \times \Gamma \times \{R, L\}$, $q_s \in Q$ is the starting state, $F \subseteq Q$ is a set of accepting and rejecting states, $*$ is a t-norm, $*^{'}$ is the dual t-conorm of $*$ and $\eta : Q \times \Gamma \times Q \times \Gamma \times \{R, L\} \to [0,1]$ is a function.*

Theorem 2.7. [11] *Let $t(n)$ be a function, where $t(n) \geq n$. Then every $t(n)$ time non-deterministic single-tape Turing machine has an equivalent $2^{O(t(n))}$ time deterministic single-tape Turing machine.*

3 Corrections in Some Previous Definitions of FTMs

In this section we explain some computational objections about the definitions of fuzzy Turing machines in [1, 8, 13]. The Wiedermann's idea in defining a Fuzzy Turing Machine (FTM) is to establish an uncertainty degree for the acceptance of a given string or the membership degree of a string to the language of an FTM. In order to compute this degree, he applies a composition on a t-norm evaluation. He considers the class of FTMs as a fuzzy extension of non-deterministic Turing machines (NTMs), where each transition has a membership degree. In order to define the acceptance degree of

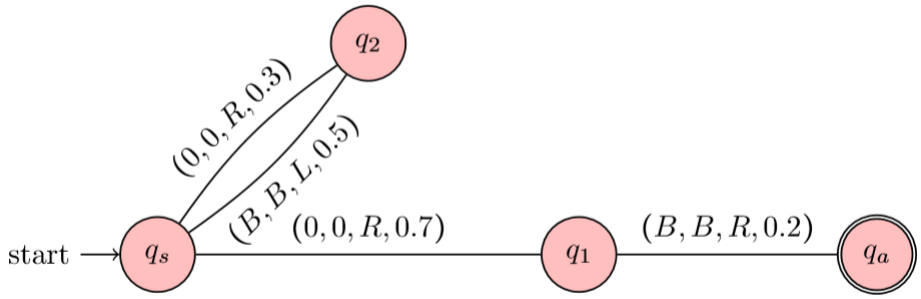


Figure 1:

an input w, he defines the degree $d(C_t)$ of an arbitrary configuration C_t, as the *maximum* degree of all computational path $C_0 \preceq C_1 \preceq^{r_1} ... \preceq^{r_t} C_t$ leading from the initial configuration $C_0 = q_0w$ to C_t:

$$d(C_t) = max\{degree(C_0, C_1, ..., C_t) \in [0,1] : \\ C_0 \preceq C_1 \preceq^{r_1} ... \preceq^{r_t} C_t \text{ is a computational path from } C_0 \text{ to } C_t\}$$

where, $degree(C_0, C_1, ..., C_t) = r_1 * ... * r_t$ if $t \geq 1$ and $degree(C_0, C_1, ..., C_t) = 1$, if $t = 0$. He also defines the accepting degree of a string w in an FTM $\mathcal{F} = (\mathcal{T}, *, \eta)$ as follows:

$$deg_{\mathcal{F}}(w) = max\{d(uq_fv) \in [0,1] : q_0w \preceq^* uq_fv \text{ for some } q_f \in F\}$$

where, $\mathcal{T} = (Q, \Sigma, \Gamma, \Delta, q_s, F)$ is a non-deterministic Turing machine and F contains only *accept* states. Note that $q_0w \preceq^* uq_fv$ denotes that there is a computational path from initial configuration q_0w to final configuration uq_fv. In [1, 8], the same setting is followed. Bedregal and Figueira [1] change the *maximum* to *supremum* in definitions of $d(C_t)$ and $deg_{\mathcal{F}}(w)$. Moniri [8] assumes that F may contain both *accept* and *reject* states. Analogously, he defines accepting and rejecting degrees of a string w using a composition on a t-conorm $*^{'}$ instead of *maximum*, where $*^{'}$ is the dual of the t-norm $*$.

Here, we perform some corrections in definitions above. For simplicity, we only discuss the case of accepting degrees and our discussion can be extended to the case of rejecting degrees naturally. We use the notation $deg_{\mathcal{F}}(w)$ for accepting degree of w by machine $\mathcal{F}$.

The important point is that *none of the definitions above for accepting degree of an input is not computationally well defined.*

; Consider the FTM shown in Figure **??** with *Product* t-norm as its t-norm. In the following there are some of its computational paths on input $w = 0$:

P_1: $q_s0 \preceq^{0.7} 0q_1B \preceq^{0.2} 0Bq_aB$

P_2: $q_s0 \preceq^{0.3} 0q_2B \preceq^{0.5} q_s0 \preceq^{0.7} 0q_1B \preceq^{0.2} 0Bq_aB$

P_3: $q_s0 \preceq^{0.3} 0q_2B \preceq^{0.5} q_s0 \preceq^{0.3} 0q_2B \preceq^{0.5} q_s0 \preceq^{0.7} 0q_1B \preceq^{0.2} 0Bq_aB$

P_4: $q_s0 \preceq^{0.3} 0q_2B \preceq^{0.5} q_s0 \preceq^{0.3} 0q_2B \preceq^{0.5} q_s0 \preceq^{0.3} \ldots$

The degrees of accepting paths P_1, P_2 and P_3 are 0.14, 0.021 and 0.00315, respectively. It is clear that there are infinitely many accepting path on input $w = 0$ leading to the configuration $0Bq_aB$. By definitions above, in order to compute the degree of configuration $0Bq_aB$, we must consider all paths leading to $0Bq_aB$. So, there is an infinite process which the machine continues searching in computational tree forever to find all accepting configurations, and so does not obtain a *computational* value for degree of $0Bq_aB$. Also note that there is an infinite loop branch P_4 on input w which cause the machine run forever, and this also prevents the process to halt.

In order to resolve this problem, we make some modifications in previous definitions of [1, 8, 13]. We take Moniri's definition of $deg_{\mathcal{F}}(w)$ as the base and make/apply the following modifications/facts:

- Any non-deterministic Turing machine can be simulated by a deterministic Turing machine using a BFS search method (See Theorem 3.16 of [11]). We will modify the introduced BFS's search method.

- We find an upper bound on the number of levels (in our modified BFS-based search) that are needed to be traversed to guarantee the existence of $deg_{\mathcal{F}}(w)$,
- We define path independent degree of a configuration just for *accepting* and *rejecting* configurations.
- We assume that the function $\eta : Q \times \Gamma \times Q \times \Gamma \times \{R, L, S\} \to [0, 1]$, gives values 0 and 1 just in the following special cases:
  (i) a direct transition from initial state to an accept or a reject state is allowed to take degree 0,
  (2) transitions *into* or *out from* an indeterminacy state are allowed to take degree 1.

In the following the modifications above are explained by detail.

Suppose that $\mathcal{F} = (\mathcal{T}, *, \eta)$ is an FTM, where $\mathcal{T} = (Q, \Sigma, \Gamma, \Delta, q_s, F)$ is a non-deterministic Turing machine. Let $\mathcal{D}$ be the deterministic Turing machine which simulates $\mathcal{T}$ using the BFS search method described in Theorem 3.16 of [11]. We make the following modifications in the introduced BFS search method:

(i) If an accepting configuration is reached in BFS search, then we prune its following branches not to traverse the following configurations initiated from that node. This modification is important because for example if we have silent moves from accept state to itself, i.e. for each $a \in \Gamma$ we have $(q_{accept}, a, q_{accept}, a, S) \in \Delta$, then we have a branch consisting infinitely many accepting configurations which are not effective in computing $deg_{\mathcal{F}}(w)$. Note that by definition of path's degree, traversing along a path will decrease the degree of path.

(ii) Assume that we are at level n of computational tree when $\mathcal{D}$ is simulating $\mathcal{T}$ on an input w. If we visit an accepting configuration in this level, then we can find an upper bound on the number of levels that are enough to be traversed (at most) to find another accepting configuration (if exists). If we could not find another accepting configuration in this level bound, then it is guaranteed that there is no other accepting configuration and we can halt and compute $deg_{\mathcal{F}}(w)$.

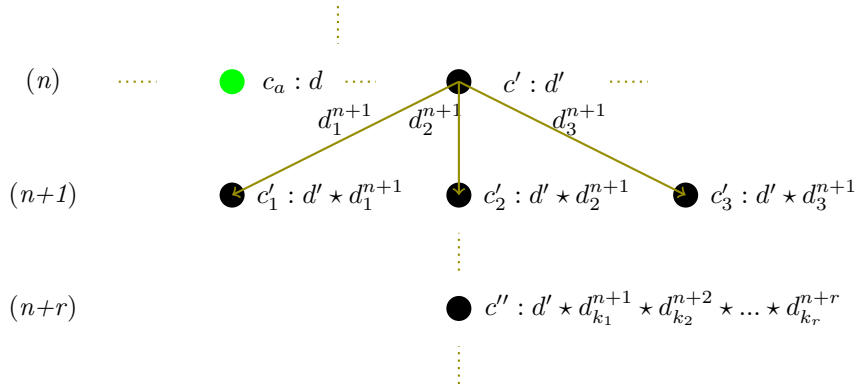
In the sequel, we explain the construction of the level's upper bound. If there exists an accepting configuration in a level of computational tree, then corresponding to each configuration in that level other than that specific accepting configuration, we obtain a *local* upper bound. The local upper bound gives the number of levels which at most are needed to be traversed in the following branches initiated from that corresponding configuration for finding other accepting configurations. Finally, we fix the desired upper bound as the maximum of these local upper bounds.

First, we mention that in the definition of the degree of a given path, a t-norm is applied between degrees of transitions along the path. So, if the degrees of transitions is strictly less than 1, then a path's degree can not increase when we move from a level to next one. Note that a path's degree strictly decrease when t-norm is not the Gödel t-norm.

Also, mention that the accepting degree of an input is defined as the *maximum* degree over all accepting configurations in the computational tree. Therefore, if we visit an accepting configuration with degree d in a level, then we traverse next levels for finding other accepting configurations which their degrees are not lower than d. In order to formalize this idea, suppose that we reach an accepting configuration c_a with degree d in a level n. Then, for each configuration $c' \neq c_a$ with degree $d^{'}$ in that level, there exist greatest natural numbers i and j such that $d^{'} * k^i \geq d$ and $d^{'} * m^j \geq d$, where k and m are non-trivial maximum and minimum values in the range of η, respectively. Note that in definitions of fuzzy-type Turing machines, Δ is a finite set and so the image of η over Δ is a finite set, so i and j exist.

Thus, since $0 \leq k, m \leq 1$, we have $i \leq log_k(d/d^{'})$ and $j \leq log_m(d/d^{'})$. Put $l_{c'} = max\{i, j\}$ as the *local bound* corresponding to the configuration c'. If we apply the same process for each arbitrary configuration in level n, then the desired upper bound $s^n_{c_a}$ corresponding to configuration c_a in level n is considered as the maximum of all these local bounds $l_{c'}$.

In the following figure, a part of computational tree between levels n and $n + r$ is shown with the exact degrees of transitions d^{n+u}_v:



The following theorem is useful for computing the accepting degree of an input by an FTM.

Theorem 3.1. *Let $\mathcal{F} = (\mathcal{T}, *, \eta)$ be an FTM. If c_a is an accepting configuration in level n of the computational tree of $\mathcal{F}$ on an input w, then it is enough to search for new accepting configurations in at most next $s^n_{c_a}$ levels.*

Proof. In above discussion, in order to find the local upper bound $l_{c'}$ corresponding to configuration c', we fixed greatest natural numbers i and j such that $d^{'} * k^i \geq d$ and $d^{'} * m^j \geq d$. In this inequalities we respectfully consider non-trivial maximum and minimum degrees k and m of transitions to cover the range of all possible degrees of transitions. So, degrees of configurations following c' are greater than d (at most) up to level $n + l_{c'}$ and obviously configurations after level $n + l_{c'}$ have degrees lower than d, so are not effective in computing $deg_{\mathcal{F}}(w)$. Therefore, if we see an accepting configuration c_a in level n, then it is enough to traverse configurations in next $s^n_{c_a}$ levels, so the proof is completed. □

Similar to the above discussion we can find an level's upper bound $t^n_{c_r}$ corresponding to a rejecting configuration c_r in level n of computational tree. In the following we propose a theorem similar to the Theorem 3.1.

Theorem 3.2. *Let $\mathcal{F} = (\mathcal{T}, *, \eta)$ be an FTM. If c_r is a rejecting configuration in level n of the computational tree of $\mathcal{F}$ on an input w, then it is enough to search for new rejecting configurations in at most next $t^n_{c_r}$ levels.*

Proof. The proof is similar to the proof of Theorem 3.1. □

Consequently, we modify previous definitions of FTMs or GFTMs to compute the accepting or rejecting degrees of inputs in an algorithmic way. In Section 4, we apply the modifications so far in defining the class of Extended Fuzzy Turing Machines.

Remark 3.3. *If the computational tree of an FTM (or a GTFM) on an input w has no loop branch, then the accepting degree (and also rejecting degree) of w can be defined computationally.*

Remark 3.4. *If in an FTM $\mathcal{E} = (\mathcal{T}, *, \eta)$ or similarly in a GFTM, the machine $\mathcal{T}$ be a deterministic Turing machine, then definitions of accepting or rejecting degrees in*[1, 8, 13] *are computationally well-defined.*

4 Extended Fuzzy Turing Machines and Some Computability Results

In this section, we extend FTMs and GFTMs to machines with some new type of states that we call *indeterminacy* states. We also study some computability properties of these extended machines.

4.1 Extended Fuzzy Turing Machines

Considering the modifications mentioned in Section 3, we introduce the class of Extended Fuzzy Turing Machines as an extension of the class of GFTMs defined by Moniri [8]. In defining extended fuzzy Turing machines, we consider some modifications with respect to Moniri's definition: (1) we specify *indeterminacy* states as a new type of states; (2) unlike Moniri, we assume that the transition relation Δ is a ***crisp*** set and it is not ***fuzzy***. If Δ be a fuzzy subset, then η has not the desired meaning.

Definition 4.1. *An Extended Fuzzy Turing Machine (briefly, EFTM) is a tuple $\mathcal{E} = (\mathcal{T}, *, *^{'}, \eta)$, where $\mathcal{T} = (Q, \Sigma, \Gamma, \Delta, q_s, F)$ is a non-deterministic Turing machine. Q is a set of states which consists a special set $\mathcal{I}$ of indeterminacy states, Σ is a set of input symbols, Γ is a set of tape symbols containing the blank symbol, Δ is a crisp subset of $Q \times \Gamma \times Q \times \Gamma \times \{R, L, S\}$, $q_s \in Q$ is the start state, $F = \mathcal{A} \cup \mathcal{R} \subseteq Q$ consists accepting and rejecting states, $*$ is a t-norm and $*^{'}$ is its dual t-conorm and $\eta : Q \times \Gamma \times Q \times \Gamma \times \{R, L, S\} \to [0, 1]$ is a function which corresponds a truth degree to each move in Δ such that:*

(*i*) *transitions that exit from or lead to an indeterminacy state, do not change the head position and the tape's content, i.e. for each transition* $(q,a,p,b,m) \in Q\times\Gamma\times Q\times\Gamma\times\{R,L,S\}$*:*

$$q\in\mathcal{I} \text{ or } \sqrt{}\in\mathcal{I} \quad\Longrightarrow\quad \dashv=\lfloor \text{ and } \Updownarrow=\mathcal{S},$$

(*ii*) *for each* $(q,a,p,b,m)\in Q\times\Gamma\times Q\times\Gamma\times\{R,L,S\}$*:*

$$\eta(q,a,p,b,m)=1 \quad \textit{iff} \quad \textit{at least one of } q \textit{ or } p \textit{ is in } \mathcal{I},$$

(*iii*) *a direct transition from initial state to an accept or a reject state can take degree 0. Also transitions from halting states to indeterminacy state take degree 0, i.e. for each* $(q,a,p,b,m)\in Q\times\Gamma\times Q\times\Gamma\times\{R,L,S\}$*:*

$$\eta(q,a,p,b,m)=0 \quad \textit{if} \quad q=q_s \textit{ and } (p=q_a \textit{ or } p=q_r)$$

$$\eta(q,a,p,b,m)=0 \quad \textit{if} \quad (q=q_a \textit{ or } q=q_r) \textit{ and } p=q_I$$

where, q_a, q_r *and* q_I *are accept, reject and indeterminacy states, respectively.*

By last condition of (*iii*)*, it results that if an EFTM halts on an input, then its indeterminacy degree would be 0.*

For EFTMs, *instantaneous description* (or configuration) which is the unique description of a machine's tape, is defined as usual (see [12]). If $\alpha,\alpha^{'}$ are two configurations, then $\alpha\preceq^{r}\alpha^{'}$ means that there is a move in Δ with truth degree r leading from α to $\alpha^{'}$ in one step. Let α_t be reachable from α_0 in t steps through the computational path $\alpha_0\preceq^{r_1}\alpha_1\preceq^{r_2}...\preceq^{r_t}\alpha_t$, then truth degree of this path is defined as $r_1*...*r_t$, if $t\geq 1$ and is defined as 1, if $t=0$. Due to non-determinism, an arbitrary configuration such as α_t can be reached from α_0 through different computational paths, and so the degree of a configuration should be defined path independently. We define the path independent degree of a configuration just for *accepting* and *rejecting* configurations. Let α_h be an accepting or a rejecting configuration, then we define its degree as follows:

$$d(\alpha_h)=*^{'}_{j\in J}a_j$$

where, $\{a_j : j\in J\}$ is the set of all truth degrees of all computational paths leading from the initial configuration α_0 to α_h. In this definition the computational paths are traversed using our modified BFS search and applying the level's upper bounds obtained in Theorems 3.1 and 3.2.

We call the computational path $\alpha_0\preceq^{r_1}\alpha_1\preceq^{r_2}...\preceq^{r_t}\alpha_t$, an accepting, a rejecting or an indeterminacy path, if α_t is an accepting, a rejecting or an indeterminacy configuration, i.e. in the form uq_av, or uq_rv or uq_iv, where u, v are two strings and q_a, q_r, q_i are states in $\mathcal{A},\mathcal{R},\mathcal{I}$, respectively.

Definition 4.2. *Let* $\mathcal{E}=(\mathcal{T},*,*^{'},\eta)$ *be an EFTM and* $w\in\Sigma^{*}_{\mathcal{T}}$*. If there exists at least one accepting (or rejecting) path on input* w*, then the accepting (or rejecting) degree of* w*, denoted by* $e_{\mathcal{E}}(w)$ *(or* $e^{'}_{\mathcal{E}}(w)$*), is defined as follows:*

$$e_{\mathcal{E}}(w)=max_{\alpha_a}\{d(\alpha_a)\mid q_sw\preceq^{*}\alpha_a\}$$

$$(e^{'}_{\mathcal{E}}(w)=max_{\alpha_r}\{d(\alpha_r)\mid q_sw\preceq^{*}\alpha_r\})$$

where, α_a *(or* α_r*) is an accepting (or a rejecting) configuration. Otherwise, if there is no desired path we define* $e_{\mathcal{E}}(w)=0(=e^{'}_{\mathcal{E}}(w))$*.*

Above, we defined path independent degree for an accepting or a rejecting configuration in a computational manner. Now, we define the path independent degree of an *indeterminacy* configuration α_i such that it is not computationally definable in general. Let α_i be an indeterminacy configuration on w, then we define:

$$d(\alpha_i)=Inf_{j\in I}b_j$$

where, for each j, b_j is the degree of a computational path leading from α_0 to α_i. Note that $d(\alpha_i)=Inf_{j\in I}b_j$ is *mathematically* definable. If there exists at least one indeterminacy path on an input w, then we define the indeterminacy degree of w as follows:

$$e^{''}_{\mathcal{E}}(w)=Inf_{\alpha_i}\{d(\alpha_i)\mid q_sw\preceq^{*}\alpha_i\}.$$

where, α_i is an indeterminacy configuration, and otherwise, $e^{''}_{\mathcal{E}}(w)=0$.

Note that if the machine $\mathcal{E}$ goes to an *infinite configuration expansion* on input w, then there are infinitely many different indeterminacy configuration α_is. Thus in defining $e^{''}_{\mathcal{E}}(w)$, we use *Infimum* instead of *minimum* on all $d(\alpha_i)s$

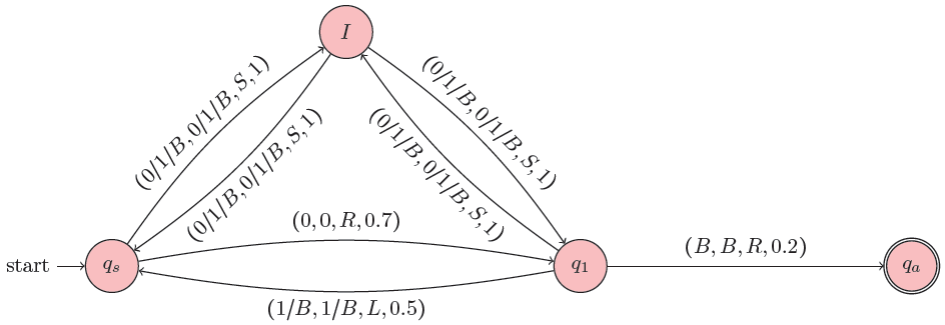


Figure 2:

Example 4.3. *Consider the EFTM $\mathcal{E}$ shown in Figure 2 with Product t-norm as its t-norm. It can be easily checked that the accepting degree of $w_1 = 0$ equals 0.14. $\mathcal{E}$ on input $w_2 = 01$ has the following infinite many sequences of indeterminacy paths with degrees 1, 0.7, 0.35, 0.245, 0.1225,..., respectively:*

P_1: $q_s01 \preceq^1 q_I01$

P_2: $q_s01 \preceq^{0.7} 0q_11 \preceq^1 0q_I1$

P_3: $q_s01 \preceq^{0.7} 0q_11 \preceq^{0.5} q_s01 \preceq^1 q_I01$

P_4: $q_s01 \preceq^{0.7} 0q_11 \preceq^{0.5} q_s01 \preceq^{0.7} 0q_11 \preceq^1 0q_I1$

$\vdots$

Hence, it is clear that $d(q_I01) = Inf_i\ d(P_i) = 0$ and so the indeterminacy degree of $w_2 = 01$ is 0.

In the sequel, we give a proposition which claims that *non-halting* of a classical Turing machine on an input is recognizable using an appropriate EFTM.

Theorem 4.4. *Let M be a classical Turing machine which loops on an input w. Then, there exists an EFTM $\mathcal{K} = (\mathcal{T}, *, *', \eta)$ whose t-norm $*$ is not Gödel t-norm, such that $e''_{\mathcal{K}}(w) = 0$.*

Proof. First, note that if $*$ is the Gödel t-norm, then EFTM shown in Figure 2 on input 01 is a counter example. It is known that for each arbitrary t-norm $*$ and all $x, y \in (0, 1)$, we have $x * y < x *_G y$, where $*_G$ is the Gödel t-norm (See [5]).

Assume that $M = (Q_M, \Sigma_M, \Gamma_M, \delta_M, q_0, F)$ is a classical Turing machine that *loops* on an input w. We construct the desired machine $\mathcal{K}$ from M as follows:

- consider the start state q_0 of M as its start state,
- correspond a non-trivial degree in open interval (0,1) to each transition of M,
- add a state q_I as indeterminacy state and consider two sided transitions between q_I and all non-accept and non-reject states of M with degree 1 and silent move for each $a \in \Gamma_M$.

By a similar discussion as in Example 4.3, it can be easily checked that if the machine M loops on an input w, then the indeterminacy degree of w in $\mathcal{K}$ is 0, i.e. $e''_{\mathcal{K}}(w) = 0$. □

Remark 4.5. *Consider the EFTM illustrated in Figure 3 with Product t-norm as its t-norm. The indeterminacy degree of $w = 0$ equals to 0, but the machine accepts $w = 0$ with degree 0.14. Therefore, the converse of Theorem 4.4 does not hold.*

Remark 4.6. *Although by Theorem 4.4, EFTMs can recognize the loops of classical Turing machines on their inputs, but we will see in Theorem 4.13 that there is no universal EFTM and so EFTMs are unable to solve the halting problem.*

Remark 4.7. *Let $\mathcal{F} = (\mathcal{T}, *, *', \eta)$ be an EFTM. If the non-deterministic Turing machine $\mathcal{T}$ has time complexity $O(t(n))$, then we do not apply our BFS search method. Therefore, time complexity of $\mathcal{F}$ in computing accepting or rejecting degree of an input is $O(t(n))$. Also note that calculating indeterminacy degree of an input is an infinite process and this degree can not be obtained computationally.*

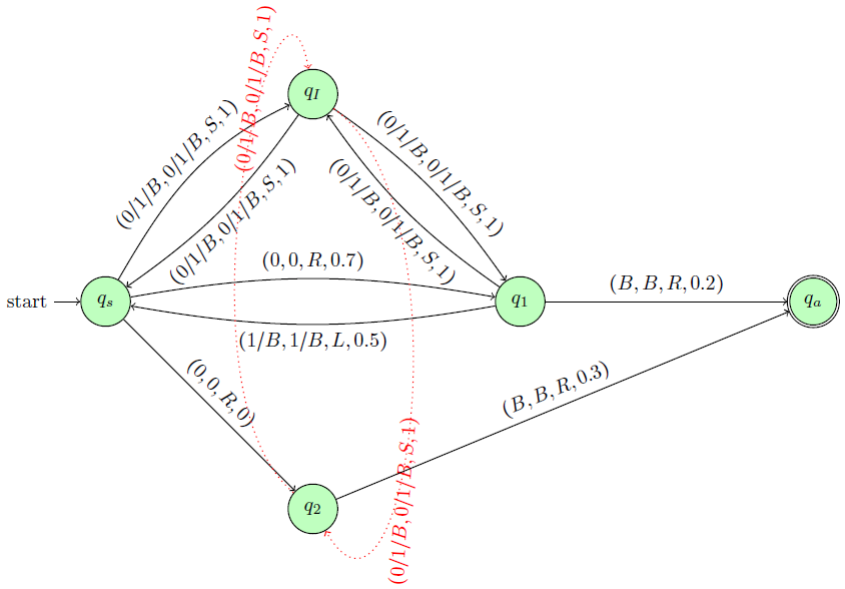


Figure 3:

Proposition 4.8. *Let $\mathcal{F} = (\mathcal{T}, *, *', \eta)$ be an EFTM. If $\mathcal{T}$ loops on some inputs, then there is function $r(n)$ such that time complexity of $\mathcal{F}$ in computing accepting or rejecting degree of an input is $2^{O(r(n))}$, where $n = |w|$.*

Proof. If $\mathcal{T}$ loops on some inputs, then we use our BFS-based search method and apply upper bounds obtained in Theorems 3.1 and 3.2. Suppose that w is an input which $\mathcal{T}$ halts on it, then let $r(n)$ be the level's number of last accepting or rejecting configuration that is visited in our search method. Hence, by a similar discussion as in the proof of Theorem 2.7 (See Theorem 7.11 of [11]), the accepting or rejecting degree of an input w can be computed in time $2^{O(r(n))}$. □

In the sequel, we define some computability notions relative to EFTMs. Let $\mathcal{E} = (\mathcal{T}, *, *', \eta)$ be an EFTM. We define $A(\mathcal{E})$, $R(\mathcal{E})$ and $I(\mathcal{E})$, as fuzzy languages *accepted*, *rejected* or *indeterminated* by $\mathcal{E}$, which their membership functions are $e_{\mathcal{E}}$, $e'_{\mathcal{E}}$ and $e''_{\mathcal{E}}$, respectively. In this setting we can think of $\mathcal{E}$ as the triple $(e_{\mathcal{E}}, e'_{\mathcal{E}}, e''_{\mathcal{E}})$. Note that $I(\mathcal{E})$ contains all pairs $(w, e''_{\mathcal{E}}(w))$ which their first element w is neither accepted nor rejected by $\mathcal{T}$. So, intuitively $I(\mathcal{E})$ consists all pairs $(w, e''_{\mathcal{E}}(w))$ which $\mathcal{T}$ loops on w.

We define the notion of an *acceptable* or a *decidable* fuzzy language similar to [8], but we define the new notion of an *indeterminable* fuzzy language as follows.

Definition 4.9. *Let L be a fuzzy language.*

(*i*) *L is acceptable if there is an EFTM $\mathcal{E}$ such that $L = A(\mathcal{E})$,*

(*ii*) *L is decidable if there is an EFTM $\mathcal{E}$ such that $L = A(\mathcal{E})$ and $L^c = R(\mathcal{E})$,*

(*iii*) *L is indeterminable if there is an EFTM $\mathcal{E}$ such that $L = I(\mathcal{E})$.*

Note that L^c is defined as $L^c(x) = 1 - L(x)$.

Example 4.10. *The following language is acceptable by the EFTM $\mathcal{E}$ illustrated in Figure 2:*

$$L = \{(0^n, (0.7)^n.0.2) : \ n \in \mathcal{N} \cup \{0\}\}$$

At the following some accepting paths on inputs $w_1 = 0$, $w_2 = 00$ and $w_3 = 000$ are given, respectively:

$w_1 = 0:$ $q_s0 \preceq^{0.7} 0q_1B \preceq^{0.2} 0Bq_aB$,

$w_2 = 00:$ $q_s00 \preceq^{0.7} 0q_10 \preceq^{1} 0q_I0 \preceq^{1} 0q_s0 \preceq^{0.7} 00q_1B \preceq^{0.2} 00Bq_aB$,

$w_3 = 000:$ $q_s000 \preceq^{0.7} 0q_100 \preceq^{1} 0q_I00 \preceq^{1} 0q_s00 \preceq^{0.7} 00q_10 \preceq^{1} 00q_I0 \preceq^{1} 00q_s0 \preceq^{0.7} 000q_1B \preceq^{0.2} 000Bq_aB$.

Example 4.11. *Complement of language $L \subseteq \{0,1\}^*$ in Example 4.10, is not acceptable by $\mathcal{E}$, but it is indeterminable by $\mathcal{E}$. The important thing is that L^c contains all binary inputs that $\mathcal{E}$ loops on them and their indeterminacy degrees are 0:*

$$L^c = \{(w,0) : \neg(\exists n \in \mathcal{N})(w = 0^n)\}$$

It can be easily checked that L^c contains all inputs of the form $1v$ and 0^n1u, where $v,u \in \{0,1\}^*$, and $\mathcal{E}$ loops on them. Thus, $L^c = I(\mathcal{E})$.

Now we compare the computational power of EFTMS to classical Turing machines. Concerning Church-Turing thesis, this comparison is an important issue in computability theory. In the following it is shown that there is a classical Turing machine which can simulate each EFTM and so, EFTMs and classical Turing machines have the same computational power.

Theorem 4.12. *There exists a classical Turing machine which can simulate each EFTM on an input w and yields the accepting, rejecting and indeterminacy degrees of w.*

Proof. We give a sketch of proof. Let $\mathcal{E} = (\mathcal{T}, *, *', \eta)$ be an EFTM, where $\mathcal{T} = (Q, \Sigma, \Gamma, \Delta, q_s, F)$ is a non-deterministic Turing machine. The machine $\mathcal{E}$ on each $w \in \Sigma^*$, outputs $(e_{\mathcal{E}}(w), e'_{\mathcal{E}}(w), e''_{\mathcal{E}}(w))$. Assume that D is the deterministic Turing machine which simulates $\mathcal{T}$ by applying our modified BFS search. We propose a classical multi-tapes Turing machine M which uses D and simulates the machine $\mathcal{E}$ such that for each input w:

(*i*) $M(w)\downarrow$ iff $\mathcal{T}(w)\downarrow$,

(*ii*) If $\mathcal{T}$ accepts w, then M outputs $(e_{\mathcal{E}}(w), 0, 0)$

(*iii*) If $\mathcal{T}$ rejects w, then M outputs $(0, e'_{\mathcal{E}}(w), 0)$,

(*iv*) If $\mathcal{T}$ does not halt on w, then also M does not halt. Note that if $\mathcal{T}$ loops on w, then $\mathcal{E}$ *mathematically* outputs $(0,0,0)$.

Note that in items (*ii*) and (*iii*), the indeterminacy degree $e''_{\mathcal{E}}(w)$ must be 0 by part (*iii*) of Definition 4.1.
Intuitively the machine M works as follows:

- Its first tape is the input tape which remains unchanged during the computation.
- Its second tape is the work tape and the simulation of $\mathcal{T}$ on w is executed on this tape.
- Its third tape holds both degree of the current configuration and also the degrees of visited accepting configurations.
- Its fourth tape holds both degree of the current configuration and also the degrees of visited rejecting configurations.
- Its fifth tape holds both degree of the current configuration and also the degrees of visited indeterminacy configurations.

Considering the upper bounds introduced in Theorems 3.1 and 3.2, when D halts on w after visiting all accepting/rejecting configurations, then the machine M computes and outputs the corresponding accepting/rejecting degrees of w using the contents of its third/fourth tape and also outputs 0 as indeterminacy degree. Also if D does not halt on w, then it transmit forever between indeterminacy state and some non-halting states, so the machine M does not halt. □

In classical computability theory, there is the notion of universal machine; a machine capable to simulate the behaviour of every other machine. Following the notion of universality for classical computability, we introduce the notions of *universality* and *semi-universality* for the class E containing all EFTMs. Since all the elements of each EFTM$\in E$ are finitely representable, we can assign a Gödel number to each EFTM, and obtain $(\mathcal{E}_\rangle)_{\rangle\in\mathcal{N}}$, an enumeration of E.

An universal EFTM $\mathcal{U}$ can be considered as a member of E which can simulate the behaviour of any other member of E, i.e. for each $i \in \mathbb{N}$ and each $w \in \Sigma^*$:

$$\mathcal{U}(\langle i, w\rangle) = \mathcal{E}_i(w)$$

such that following conditions hold:

(*i*) $\mathcal{E}_i(w)\downarrow$ iff $\mathcal{U}(\langle i,w\rangle)\downarrow$,

(*ii*) if $\mathcal{E}_i(w)\downarrow$ then $e_{\mathcal{E}_i}(w) = e_{\mathcal{U}}(\langle i,w\rangle), e'_{\mathcal{E}_i}(w) = e'_{\mathcal{U}}(\langle i,w\rangle)$ and $e''_{\mathcal{E}_i}(w) = e''_{\mathcal{U}}(\langle i,w\rangle)$.

Note that $\langle .,. \rangle : \mathbb{N} \times \Sigma^* \to \Sigma^*$ is the usual pairing function. The following theorem shows that unlike the classical scenario, there is no universal EFTM satisfying the above two conditions.

Theorem 4.13. *There is no universal EFTM.*

Proof. Suppose for the sake of contradiction that $\mathcal{U} = ((Q, \Sigma, \Gamma, \Delta, q_s, F), *, *^{'}, \eta)$ is the desired universal EFTM as described above. Let

$$d = max_{\sigma \in \Delta}(\{\eta(\sigma) : \eta(\sigma) < 1\} \cup \{0\})$$

Obviously, $\mathcal{U}$ has no *non-indeterminacy* computational path with degree $d^{'}$, where $d^{'} \in [0, 1]$ and $d < d^{'} < 1$. Intuitively, d is the maximal degree of all computational paths of $\mathcal{U}$ which is lower than 1, i.e. for each arbitrary sequence of possible moves $\sigma_1, ..., \sigma_n$ corresponding to a computational path on an input w, the following degree of this sequence is either 1 or lower than d:

$$\eta(\sigma_1) * ... * \eta(\sigma_n)$$

Consider a $d_0 \in [0, 1]$ such that $d < d_0 < 1$. Construct an EFTM $\mathcal{E}$ which has only a state q as its both start and accept states. Assume that for each $a \in \Sigma$, this machine has a transition (q, a, q, a, R) with degree d_0. Then, it can be easily shown that the $A(\mathcal{E}) = \{(\sqsupseteq, \lceil_{\prime}) : \sqsupseteq \in \pm^*\}$ and the language of $\mathcal{E}$ is $L = \{(w, (d_0, 0, 0)) : w \in \Sigma^*\}$. Suppose that the Gödel number of this machine is r. Clearly, $\mathcal{U}(\langle w, r\rangle) \downarrow$ iff $\mathcal{E}(w) \downarrow$ and $e_{\mathcal{U}}(\langle r, w\rangle) = e_{\mathcal{E}}(w) = d_0, e_{\mathcal{U}}^{'}(\langle r, w\rangle) = 0$ and $e_{\mathcal{U}}^{''}(\langle r, w\rangle) = 0$, which is impossible. Therefore, there is no universal EFTM. □

Now we define a new notion *Semi-Universality* that is weaker than Universality, and show that there is a semi-universal EFTM. A *Semi-Universal* machine is an EFTM $\mathcal{U}'$ (regarded as an indeterminator) with the ability to simulate the behaviour of any other EFTM in E as in the following way, such that for each $i \in \mathbb{N}$ and each $w \in \Sigma^*$:

(*i*) $\mathcal{U}'(\langle i, w\rangle) \downarrow$ if and only if $\mathcal{E}_i(w) \downarrow$,

(*ii*) if $\mathcal{E}_i(w) \uparrow$ then $e_{\mathcal{U}'}^{''}(\langle i, w\rangle) = 0$.

Proposition 4.14. *There is a semi-universal EFTM.*

Proof. Assume that $i \in \mathbb{N}$ and $w \in \Sigma^*$. The machine $\mathcal{U}'$ decodes i to obtain the EFTM $\mathcal{E}_i$, then as the process proposed in the proof of Theorem 4.4, construct the machine $\mathcal{K}_{(\mathcal{E}_\rangle, \sqsupseteq)}$ and outputs as the same as $\mathcal{K}_{(\mathcal{E}_\rangle, \sqsupseteq)}$ outputs. Hence,

(*i*) $\mathcal{U}'(\langle i, w\rangle) \downarrow \iff \mathcal{K}_{(\mathcal{E}_\rangle, \sqsupseteq)}(\sqsupseteq) \downarrow \iff \mathcal{E}_\rangle(\sqsupseteq) \downarrow$,

(*ii*) if $\mathcal{E}_i(w) \uparrow$ then $e_{\mathcal{K}_{(\mathcal{E}_\rangle, \sqsupseteq)}}^{''}(w) = 0 = e_{\mathcal{U}'}^{''}(\langle i, w\rangle)$.

Therefore, $\mathcal{U}$ satisfies the desired properties and is a semi-universal EFTM. □

Proposition 4.15. *The following language ${A_{TM}}^c$ is indeterminable.*

$${A_{TM}}^c = \{((M, w), 0) : M \text{ is a classical Turing machine that accepts } w\}$$

Proof. For each (M, w), run the semi-universal machine $\mathcal{U}'$ on (M, w). If $M(w) \downarrow$, then $\mathcal{K}_{(\mathcal{M}, \sqsupseteq)}(\sqsupseteq) \downarrow$ and so $\mathcal{U}'(\langle M, w\rangle) \downarrow$. Otherwise, if $M(w) \uparrow$, then $e_{\mathcal{K}_{(\mathcal{M}, \sqsupseteq)}}^{''}(w) = e_{\mathcal{U}'}^{''}(\langle M, w\rangle) = 0$. Therefore, ${A_{TM}}^c = I(\mathcal{U}')$ and so ${A_{TM}}^c$ is indeterminable by $\mathcal{U}$. □

Proposition 4.16. *The following language $Halt^c$ is indeterminable.*

$$Halt^c = \{((M, w), 0) : M \text{ is a classical Turing machine that halts on } w\}$$

Proof. The Proof is similar to the proof of Proposition 4.15. □

In the following we restate Proposition 3.5, Corollary 3.6 and Proposition 3.8 of [8], which also hold here.

Proposition 4.17. *A fuzzy language L is decidable if and only if L and L^c are acceptable.*

Corollary 4.18. *If a fuzzy language L is decidable, then L^c is decidable,*

Proposition 4.19. *Let L_1 and L_2 be two fuzzy languages. Assume that L_1 and L_2 are accepted by EFTMs $\mathcal{E}_1$ and $\mathcal{E}_2$ equipped with the same t-norm $*$ and let $*^{'}$ be the dual t-conorm of $*$. Then $L_1 *^{'} L_2$ is accepted by an EFTM equipped with the same t-norm and t-conorm.*

Finally, we restate Propositions 4.1. and 4.2 of [8], which propose some correspondences between EFTMs and the class of classical r.e. and co-r.e. languages.

Proposition 4.20. *Let L be a classical r.e. language. Then there is an EFTM $\mathcal{E}$ such that for each arbitrary input w, we have $w \in L$ if and only if $\mathcal{E}$ accepts w with a non-zero degree b, rejects it with degree 0 and indeterminates it with degree 0.*

Proof. The proof is similar to the proof of Propositions 4.1. of [8]. Suppose that M is a classical Turing machine which recognizes L. Without loss of generality we assume that M has only accepting states as its final states and eventually accepts an input w if w is in L and never halts on other inputs. Let t be a real number in open interval (0,1). Construct the EFTM $\mathcal{E}$ as follows:

- change the starting state of M to a non-starting state q,
- correspond degree t to all transitions of M,
- consider a starting state q_s,
- consider two nondeterministic transitions from q_s to:
  (i) state q of the new modified version of M with degree t,
  (ii) a new rejecting state with degree 0 (note that here the degree 0 is allowed here by Definition 4.1),
- consider transitions from all non-accept and non-reject states to an indeterminacy state with degree 1 and vice versa,
- consider transitions from all accept and reject states to the indeterminacy state with degree 0.

It is obvious that for each input w if $w \in L$, then there is a number $b \in (0,1)$ such that $e_{\mathcal{E}}(w) = b \leq t$ and $e'_{\mathcal{E}}(w) = e''_{\mathcal{E}}(w) = 0$. Therefore, $\mathcal{E}$ is the desired machine. □

Proposition 4.21. *Let L be a (classical) co-r.e. language. There is an EFTM $\mathcal{E}$ such that for each input w, we have $w \notin L$ if and only if $\mathcal{E}$ rejects w with a non-zero degree r, accepts it with degree 0 and indeterminates it with degree 0.*

Proof. The proof is based on the proof of Propositions 4.2 in [8] and is similar to the proof of Proposition 4.20. □

5 Conclusions

We modified some previous definitions of fuzzy-type Turing machines. We applied a BFS-based search method and obtained some level's upper bounds to calculate degrees of accepting and rejecting in a computational sense. We proposed the class of Extended Fuzzy Turing Machines that are equipped with indeterminacy states. It was shown that these machines are useful in identifying the loops of classical Turing machines. We also defined the notion of indeterminable language and showed that the complement of traditional halting problem is an indeterminable language. Afterwards, we proved that there is not any universal extended machine, but there is a semi-universal extended machine. We showed that our extended machines and classical Turing machines have the same computational power; so the extended machines can not solve the halting problem.

Acknowledgement

Author would like to thank anonymous referees for their useful comments. Also, I would like to thank M. R. Bahrami for some comments.

References

[1] B. C. Bedregal, S. Figueira, *On the computing power of fuzzy Turing machines*, Fuzzy Sets and Systems, **159** (2008), 1072–1083.

[2] L. Biacino, G. Gerla, *Recursively enumerable L-sets*, Mathematical Logic Quarterly, **33(2)** (1987), 107–113.

[3] H. Farahani, *Meta-type Fuzzy Computations and Fuzzy Complexity*, Journal of Intelligent and Fuzzy Systems, **34** (2018), 81–92.

[4] G. Gerla, *Sharpness relation and decidable fuzzy sets*, IEEE Transactions on Automatic Control, **27(5)** (1982), 1113–1113.

[5] P. Hájek, *Metamathematics of Fuzzy Logic*, Kluwer Academic Publishers, Dordrecht, 1998.

[6] L. Harkteroad, *Fuzzy recursion. ret's arid isols*, Zeitschrift Math. Logik Grundlagen Math, **30** (1984), 425–430.

[7] E. T. Lee, L. A. Zadeh, *Note on fuzzy languages*, Information Sciences, **4(1)** (1969), 421–434.

[8] M. Moniri, *Fuzzy and Intuitionistic Fuzzy Turing Machines*, Fundamenta Informaticae, **123** (2013), 305–315.

[9] E. S. Santos, *Fuzzy algorithms*, Information and Control, **17** (1970), 326–339.

[10] E. S. Santos, *Fuzzy and probabilistic programs*, Information Sciences, **10** (1976), 331–335.

[11] M. Sipser, *Introduction to the Theory of Computation, 3rd Edition*, Cengage Learning, Boston, MA, 2012.

[12] J. Wiedermann, *Charactrizing the super-Turing power and efficiency of classical fuzzy Turing machines*, Theoretical Computer Science, **317** (2004), 61–69.

[13] J. Wiedermann, *Fuzzy Turing machines revised*, Computer Artificial Intelligence, **21(3)** (2003), 1–13.

[14] L. A. Zadeh, *Fuzzy algorithms*, Information and Control, **2** (1968), 94–102.

[15] L. A. Zadeh, *Fuzzy sets*, Information and Control, **8** (1965), 338–353.

SOME IMPROVEMENTS IN FUZZY TURING MACHINES

H. FARAHANI

بهبودهایی در ماشین‌های تورینگ فازی

چکیده. در این مقاله برخی تعریف‌های پیشین ماشین‌های تورینگ فازی را به منظور یافتن درجه‌های پذیرش و رد به شکلی محاسباتی بهبود می‌دهیم. یک روش جستجوی بر اساس BFS و برخی کران‌های بالای سطح را به‌کار می‌گیریم تا یک فرآیند محاسباتی در محاسبه درجه‌های پذیرش و رد را معرفی نماییم. سپس کلاس ماشین‌های تورینگ فازی تعمیم‌یافته که مجهز به حالت‌های نامشخص هستند، را معرفی می‌کنیم. این حالت‌های نامشخص در یک فرآیند ریاضی در مشخص کردن حلقه‌های نامتناهی ماشین‌های تورینگ کلاسیک به‌کار می‌روند. در ادامه پس از تعریف زبان‌های پذیرفتنی و تصمیم‌پذیر، مفهوم جدید زبان‌های نامشخص را تعریف می‌کنیم. یک زبان نامشخص به اجراهای بدون توقف یک ماشین متناظر می‌شود. در ادامه نشان می‌دهیم که ماشین تعمیم‌یافته جهانی وجود ندارد، که از این حکم نتیجه می‌شود ماشین‌های تعمیم یافته قادر به حل مساله توقف نیستند. همچنین نشان می‌دهیم ماشین‌های تورینگ تعمیم‌یافته دارای قدرت محاسباتی یکسانی با ماشین‌های تورینگ کلاسیک هستند. سپس مفهوم جدید شبه–جهانی بودن را تعریف می‌کنیم و اثبات می‌کنیم که یک ماشین تعمیم‌یافته ی شبه–جهانی وجود دارد. این ماشین می‌تواند مکمل مساله توقف کلاسیک را نامشخص کند. به علاوه، به هر زبان r.e یا co-r.e، زبانی را نظیر می‌کنیم که با تعدادی ماشین تورینگ فازی تعمیم‌یافته مرتبط هستند.