symmetry

MDPI

Article

Neutrosophic Computability and Enumeration

Ahmet Çevik [1], Selçuk Topal [2,*] and Florentin Smarandache [3]

[1] Gendarmerie and Coast Guard Academy, Department of Science, 06805 Ankara, Turkey, a.cevik@hotmail.com
[2] Department of Mathematics, Faculty of Science and Arts, Bitlis Eren University, 13000 Bitlis, Turkey
[3] Department of Mathematics, University of New Mexico, Gallup, NM 87301, USA; smarand@unm.edu
* Correspondence: s.topal@beu.edu.tr; Tel.: +90-532-709-0239

Received: 28 October 2018; Accepted: 13 November 2018; Published: 16 November 2018

Abstract: We introduce oracle Turing machines with neutrosophic values allowed in the oracle information and then give some results when one is permitted to use neutrosophic sets and logic in relative computation. We also introduce a method to enumerate the elements of a neutrosophic subset of natural numbers.

Keywords: computability; oracle Turing machines; neutrosophic sets; neutrosophic logic; recursive enumerability; oracle computation; criterion functions

1. Introduction

In classical computability theory, algorithmic computation is modeled by Turing machines, which were introduced by Alan M. Turing [1]. A *Turing machine* is an abstract model of computation defined by a 7-tuple $(Q, \Sigma, \Gamma, \delta, q_0, F, \{L, R\})$, where Q is a finite set of states, Σ is the alphabet, Γ is the tape alphabet, $q_0 \in Q$ is the starting state, $F \subset Q$ is a set of halting states, the set $\{L, R\}$ denotes the possible left (L) and right (R) move of the tape head, and δ is the transition function, defined as:

$$\delta : Q \times \Gamma \to Q \times \Sigma \times \{L, R\}.$$

Each transition is a step of the computation. Let w be a string over the alphabet Σ. We say that a Turing machine on input w *halts* if the computation ends with some state $q \in F$. The output of the machine, in this case, is whatever was written on the tape at the end of the computation. If a Turing machine M on input w halts, then we say that M is defined on w. Since there is a one-to-one correspondence between the set of all finite strings over Σ and the set of natural numbers $\mathbb{N} = \{0, 1, 2, \ldots\}$, without loss of generality we may assume that Turing machines are defined from $\mathbb{N}$ to $\mathbb{N}$.

Standard Turing machines admit partial functions, i.e., functions that may not be defined on every input. The class of functions computable by Turing machines are called *partial recursive (computable)* functions. We shall not delve into the details about what is meant by a function or set that is computable by a Turing machine. We assume that the reader is familiar with the basic terminology. However, for a detailed account, the reader may refer to Reference [2–4]. Using a well known method called *Gödel numbering*, originated from Gödel's celebrated 1931 paper [5], it is possible to have an algorithmic enumeration of all partial recursive functions. We let Ψ_i denote the ith partial recursive function, i.e., the ith Turing machine.

If a partial recursive function is defined on every argument we say that it is *total*. Total recursive functions are simply called *recursive* or *computable*. Since there are countable infinitely many Turing machines, there are countable infinitely many computable functions. Computable sets and functions are widely used in mathematics and computer science. However, nearly all functions are non-computable.

Since there are $2^{\aleph_0}$ functions from $\mathbb{N}$ to $\mathbb{N}$ and only $\aleph_0$ many computable functions, there are uncountably many non-computable functions.

Oracle Turing machines, introduced by Alan Turing [6], are used for relativizing the computation with respect to a given set of natural numbers. An *oracle Turing machine* is a Turing machine with an extra *oracle tape* containing the characteristic function of a given set of natural numbers. The *characteristic function* of a set $S \subset \mathbb{N}$ is defined as:

$$\chi_S(x) = \begin{cases} 1 & \text{if } x \in S \\ 0 & \text{if } x \notin S. \end{cases}$$

We may also think of χ_S as an infinite binary sequence and call it the *characteristic sequence* of S. For a set $S \subset \mathbb{N}$, we let $S(i)$ denote $\chi_S(i)$. So the characteristic sequence of a set S simply gives the membership information about natural numbers regarding S. For a given an oracle Turing machine with the characteristic sequence of a set S provided in the oracle tape, functions are denoted as computable by the machine relative to the *oracle* S. If the oracle Turing machine with an oracle S computes a function f, then we say that f is *computable in* S or we say S *computes* f. We denote the ith oracle Turing machine with an oracle A by $\Psi_i(A)$. Then, it makes sense to write $\Psi_i(A) = B$ if A computes B.

Now we shall look at a non-standard Turing machine model based on neutrosophic sets. Neutrosophic logic, first introduced by Smarandache [7,8], is a generalization of classical, fuzzy and intuitionistic fuzzy logic. The key assumption of neutrosophy is that every idea not only has a certain degree of truth, as is generally taken in many-valued logic contexts, but also has degrees of falsity and indeterminacy, which need to be considered independently from each other. A neutrosophic set relies on the idea that there is a degree of probability that an element is a member of the given set, a degree that the very same element is *not* a member of the set, and a degree that the membership of the element is indeterminate for the set. For our purpose we take subsets of natural numbers. Roughly speaking, if n were a natural number and if A were a neutrosophic set, then there would be a probability distribution $p_{\in}(n) + p_{\notin}(n) + p_I(n) = 1$, where $p_{\in}(n)$ denotes the probability of n being a member of A, $p_{\notin}(n)$ denotes the probability of n not being a member of A, and $p_I(n)$ denotes the degree of probability that the membership of n is indeterminate in A. Since the probability distribution is expected to be normalized, the summation of all probabilities must be equal to unity. We should note however that the latter requirement can be modified depending on the application.

The above interpretation of a neutrosophic set can be in fact generalized to any multi-dimensional collection of attributes. That is, our attributes did not need to be merely about membership, non-membership, and indeterminacy, but it could range over any finite set of attributes $a_0, a_1, \ldots, a_k$ and $b_0, b_1, \ldots b_k$ so that the value of an element would range over (x, y, I) such that $x \in a_m$ and $y \in b_m$ for $0 \leq m \leq k$. The set of attributes can also be countably infinite or even uncountable. However, we are not concerned with these cases. We shall only consider the membership attribute discussed above.

We are particularly interested in subsets of natural numbers $A \subset \mathbb{N}$, in our study. Any neutrosophic subset A of natural numbers (we shall occasionally denote such a set by A^N) is defined in the form of ordered triplets:

$$\{\langle p_{\in}(0), p_{\notin}(0), p_I(0)\rangle, \langle p_{\in}(1), p_{\notin}(1), p_I(1)\rangle, \langle p_{\in}(2), p_{\notin}(2), p_I(2)\rangle, \ldots\},$$

where, for each $i \in \mathbb{N}$, $p_{\in}(i)$ denotes the degree of probability of i being an element of A, $p_{\notin}(i)$ denotes the probability of i being not an element of A, and $p_I(i)$ denotes the probability of i being indetermined. Since we assume a normalized probability distribution, we have that for every $i \in \mathbb{N}$:

$$p_{\in}(i) + p_{\notin}(i) + p_I(i) = 1.$$

2. Oracle Turing Machines with Neutrosophic Values

Now we can extend the notion of relativized computation based on neutrosophic sets and neutrosophic logic. For this we introduce oracle Turing machines with neutrosophic oracle tape. The general idea is as follows. Standard oracle tape contains the information of the characteristic sequence of a given set $A \subset \mathbb{N}$. We extend the definition of the characteristic function to neutrosophic sets as follows.

Definition 1. *Let $A \subset \mathbb{N}$ be a set. A neutrosophic oracle tape is a countably infinite sequence $t_0, t_1, \ldots$ where $t_i = \langle a, b, c \rangle$ is an ordered triplet and $a, b, c \in \mathbb{Q}$, so that a is the probability value of i such that $i \in A$, b is the probability value of i such that $i \notin A$, and c is the probability of i being indeterminate for A.*

The overall picture of a neutrosophic oracle tape can be seen in Figure 1. Now we need to modify the notion of the characteristic sequence accordingly.

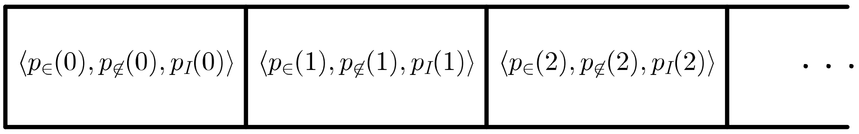
$\langle p_{\in}(0), p_{\notin}(0), p_I(0) \rangle$
$\langle p_{\in}(1), p_{\notin}(1), p_I(1) \rangle$
$\langle p_{\in}(2), p_{\notin}(2), p_I(2) \rangle$
...

Figure 1. Neutrosophic oracle tape.

Definition 2. *Let $S \subset \mathbb{N}$ be a set and let B denote the blank symbol in the alphabet of the oracle tape. The neutrosophic characteristic function of S is defined by*

$$\chi_S^N(x) = \begin{cases} \langle 1, 0, I \rangle & \text{if } p_{\in}(x) > 0 \text{ and } p_{\notin}(x) > 0 \text{ and } p_I(x) > 0 \\ \langle B, 0, I \rangle & \text{if } p_{\in}(x) = 0 \text{ and } p_{\notin}(x) > 0 \text{ and } p_I(x) > 0 \\ \langle B, B, I \rangle & \text{if } p_{\in}(x) = 0 \text{ and } p_{\notin}(x) = 0 \text{ and } p_I(x) > 0 \\ \langle B, B, B \rangle & \text{if } p_{\in}(x) = 0 \text{ and } p_{\notin}(x) = 0 \text{ and } p_I(x) = 0 \\ \langle 1, B, I \rangle & \text{if } p_{\in}(x) > 0 \text{ and } p_{\notin}(x) = 0 \text{ and } p_I(x) > 0 \\ \langle 1, B, B \rangle & \text{if } p_{\in}(x) > 0 \text{ and } p_{\notin}(x) = 0 \text{ and } p_I(x) = 0 \\ \langle 1, 0, B \rangle & \text{if } p_{\in}(x) > 0 \text{ and } p_{\notin}(x) > 0 \text{ and } p_I(x) = 0 \\ \langle B, 0, B \rangle & \text{if } p_{\in}(x) = 0 \text{ and } p_{\notin}(x) > 0 \text{ and } p_I(x) = 0 \end{cases}$$

The idea behind this definition is to label the distributions which have significant probability value with respect to a pre-determined probability threshold value, in this case we assume this value to be 0 by default. Note that this threshold value could be defined for any $r \in \mathbb{Q}$ so that instead of being greater than 0, we would require the probability for that attribute to be greater than r in order to be labelled. We will talk about the properties of defining an arbitrary threshold value and its relation to neutrosophic computations in the next section.

Definition 3. *A neutrosophic oracle Turing machine is a Turing machine with an additional neutrosophic oracle tape $(Q, \Sigma, \Gamma, \Gamma', \delta, q_0, F, \{L, R\})$, where Q is a finite set of states, Σ is the alphabet, Γ is the tape alphabet, Γ' is the neutrosophic oracle tape alphabet containing the blank symbol B, $q_0 \in Q$ is the starting state, $F \subset Q$ is a set of halting states, the set $\{L, R\}$ denotes the possible left (L) and right (R) move of the tape head, and δ is the transition function defined as:*

$$\delta : Q \times \Gamma \times \Gamma' \to Q \times \Sigma \times \{L, R\}^2.$$

Theorem 1. *Any neutrosophic oracle Turing machine can be simulated by a standard Turing machine.*

Proof. Assuming the Church–Turing thesis in the proof, we only need to argue that standard oracle tapes can in theory represent neutrosophic oracle tapes. In fact any neutrosophic oracle tapes can be represented by three standard oracle tapes each of which contains one and only one attribute. The *i*th cell of the first oracle tape contains the probability value $p_{\in}(i)$. The *i*th cell of the second oracle tape contains the value $p_{\notin}(i)$. Similarly, the *i*th cell of the third oracle tape contains $p_I(i)$.

We also need to argue that a three-tape oracle standard Turing machine can be simulated by a single tape oracle Turing machine. Let Γ be the oracle alphabet. We define an extension Γ' of Γ by introducing a delimeter symbol # to separate each attribute for a given number *i*. We define another delimiter symbol $\perp$ to separate each $i \in \mathbb{N}$. Let $\Gamma' = \Gamma \cup \{\#, \perp\}$. Then, a neutrosophic oracle tape can be represented by a single oracle tape with the tape alphabet Γ'. The oracle tape will be in the form:

$$p_{\in}(0)\#p_{\notin}(0)\#p_I(0)\perp p_{\in}(1)\#p_{\notin}(1)\#p_I(1)\perp \ldots$$

The symbol $\perp$ determines a counter for *i*, whereas for each *i*, the symbol # determines a counter for the attribute. □

A neutrosophic set A *computes* another neutrosophic set B if using finitely many pieces of information of the characteristic sequence of A determines the *i*th entry of the characteristic sequence of B given any index $i \in \mathbb{N}$. Then, based on this definition, a set $B \subset \mathbb{N}$ is *neutrosophically computable* in A if $B = \Psi_e^N(A)$ for some $e \in \mathbb{N}$, where Ψ_e^N denotes the e-th neutrosophic oracle Turing machine. If $B = \Psi_e^N(A)$ for some $e \in \mathbb{N}$, we denote this by $B \leq_N A$. If $B \leq_N A$ and $A \leq_N B$, then we say that A and B are *neutrosophically equivalent* and denote this by $A \equiv_N B$. Intuitively, $A \equiv_N B$ means that A and B are neutrosophic subsets of natural numbers, and they have the same level of neutrosophic information complexity. We leave the discussion on the properties of the equivalence classes induced by $\equiv_N$ for another study as it is beyond the scope of this paper.

3. Neutrosophic Enumeration and Criterion Functions

We now introduce the concept of neutrosophic enumeration of the members of neutrosophic subsets of natural numbers. Since we talk about enumeration, we must only take countable sets into consideration. It is known from classical computability that, given a set $A \subset \mathbb{N}$, A is called *recursively enumerable* if there exists some $e \in \mathbb{N}$ such that A is the domain of Ψ_e. We want to define the neutrosophic counterpart of this notion, but we need to be careful about the indeterminate cases, an intrinsic property in neutrosophic logic.

Definition 4. *A set A is called neutrosophic Turing enumerable if there exists some $e \in \mathbb{N}$ such that A is the domain of Ψ_e^N restricted to elements whose probability degree of membership is greater than a given probability threshold. More precisely, if $r \in \mathbb{Q}$ is a given probability threshold, then A is neutrosophic Turing enumerable if A is the domain of $\Psi_e^N(\emptyset)$ restricted to those elements i such that $p_{\in}^A(i) \geq r$, where $p_{\in}^A(i)$ denotes the degree of probability of membership of i in A.*

If the *e*th Turing machine is defined on the argument *i*, we denote this by $\Psi_e(i)\downarrow$. The *halting set* in classical computability theory is defined as:

$$K = \{e : \Psi_e(e)\downarrow\}.$$

It is known that K is recursively enumerable but not recursive. Unlike in classical Turing computability, we show that neutrosophically computable sets allow us to neutrosophically compute the halting set. The way to do this goes as follows. A single neutrosophic subset of natural numbers is not enough to compute the halting set. Instead, we take the union of all neutrosophically computable subsets of natural numbers by taking an infinite join which will code the information of the halting set.

Let $\{A_i\}_{i\in\mathbb{N}}$ be a countable sequence of subsets of $\mathbb{N}$. The *infinite join* is defined by

$$\bigoplus\{A_i\} = \{\langle i,x\rangle : x \in A_i\},$$

where $\langle i,j\rangle$ is mapped to a natural number using a uniform pairing function $\mathbb{N}\times\mathbb{N}\to\mathbb{N}$.

Theorem 2. *Let A^N be a neutrosophic subset of $\mathbb{N}$. Then, $\bigoplus\{A_i^N\} \equiv_N K$*

Proof. We first show that $\bigoplus\{A_i^N\} \geq_N K$. The infinite join of all neutrosophically computable sets computes the halting set. Let $\{\Psi_i^N\}_{i\in\mathbb{N}}$ be an effective enumeration of neutrosophic Turing functionals. Let $\{A_i^N\}$ be the corresponding neutrosophic sets, each of which is computable by Ψ_i^N. To compute K, we let $\bigoplus\{A_i^N\}$ be the infinite join of all A_i^N. To know whether $\Psi_i(i)$ is defined or not, we see if $p_\in(i) + p_\notin(i) > 0.5$. If so, then $\Psi_i(i)\downarrow$. Otherwise it must be that $p_I(i) > 0.5$. In this case, $\Psi_i(i)$ is undefined.

Next, we show $\bigoplus\{A_i^N\} \leq_N K$. To prove this, we assume that there exists an oracle for K. If $i\in K$, then there exist indices $x,y\in\mathbb{N}$ such that $\langle x,y\rangle = i$ and it must be that $p_\in(i)+p_\notin(i) > 0.5$ since $\Psi_i(i)$ is defined, but we may not know whether $i\in A_i$ or $i\notin A_i$. If $i\notin K$, then the same argument holds to prove this case as well. □

The use of the probability ratio 0.5 is for convenience. This notion will be generalized later on. Classically speaking, given a subset of natural numbers, we can easily convert it to a neutrosophic set preserving the membership information of the given classical set. Suppose that we are given a set $A\subset\mathbb{N}$ and we want to convert it to a neutrosophic set with the same characteristic sequence. The neutrosophic counterpart A^N is defined, for each $i\in\mathbb{N}$, as:

$$A^N(i) = \begin{cases} \langle i, 1-i, 0\rangle & \text{if } A(i)=1 \\ \langle 1-i, i, 0\rangle & \text{otherwise.}\end{cases}$$

We now introduce the tree representation of neutrosophic sets and give a method, using trees, to approximate its classical counterpart. Suppose that we are given a neutrosophic subset of natural numbers in the form:

$$A^N = \{\langle p_\in(i), p_\notin(i), p_I(i)\rangle\}_{i\in\mathbb{N}}.$$

We use the probability distribution to decide which element will be included in the classical counterpart. If A^N is a neutrosophic subset of natural numbers, the *classical counterpart of A^N* is defined as:

$$A(i) = \begin{cases} 1 & \text{if } p_\in(i) > p_\notin(i) \\ 0 & \text{if } p_\in(i) < p_\notin(i).\end{cases}$$

Now we introduce a simple conversion using trees. The aim is to approximate to the classical counterpart of a given neutrosophic set A^N in a computable fashion. For this we start with a full ternary tree, as given in Figure 2, coding all possible combinations.

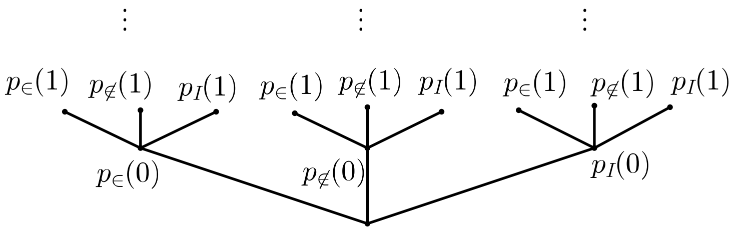
$p_\in(1)$ $p_\notin(1)$ $p_I(1)$
$p_\in(1)$ $p_\notin(1)$ $p_I(1)$
$p_\in(1)$ $p_\notin(1)$ $p_I(1)$
$p_\in(0)$
$p_\notin(0)$
$p_I(0)$

Figure 2. Approximating a neutrosophic set with a classical set through a ternary tree.

The correct interpretation of this tree is as follows. Each branch represent a possible element of the set we want to construct. For instance, if $p_{\in}(0)$ has the largest probability value among $p_{\in}(0)$, $p_{\notin}(0)$, $p_I(0)$, then we choose $p_{\in}(0)$ and define 0 to be an element of the classical set we construct. If either $p_{\notin}(0)$ or $p_I(0)$ is greater than $p_{\in}(0)$, then we know 0 is not an element of the constructed set. Since we are defining a classical set, the only time when some $i \in \mathbb{N}$ is in the constructed set is if $p_{\in}(i) > p_{\notin}(i)$ and $p_{\in}(i) > p_I(i)$. Continuing along this line, if $p_{\notin}(1)$, say, has the greatest probability value among $p_{\in}(1)$, $p_{\notin}(1)$, $p_I(1)$, then 1 will not be an element. So far, 0 is an element and 1 is not an element. So in the tree we choose the leftmost branch and then next we choose the middle branch. Repeating this procedure for every $i \in \mathbb{N}$, we end up defining a computable infinite path on this ternary tree which defines elements of the set being constructed. At each step, we simply take the maximum probability value and select that attribute. The infinite path defines a computable approximation to the classical counterpart of A^N using the tree method.

Earlier we defined Turing machines with a neutrosophic oracle tape. Suppose that the characteristic sequence of a neutrosophic set A can be considered as an oracle. Then, the e-th neutrosophic oracle Turing machine can compute a function of the same characteristic. That is, not only can Turing machines with neutrosophic oracles compute classical sets, but they can also compute neutrosophic sets. It is important to note that we need to modify the definition of standard oracle Turing machines in order to use neutrosophic sets. We add the symbol I to the alphabet of the oracle tape. The transition function δ^N is then defined as:

$$\delta^N : Q \times \Sigma \times \Gamma \to Q \times \Sigma \cup \{I\} \times \{L, R\}^2.$$

We say that the neutrosophic oracle Turing machine, say Ψ_e^N, *computes* a neutrosophic set B if $\Psi_e^N = B$.

We now turn to the problem of enumerating members of a neutrosophic subset A of natural numbers. Normally, general intuition suggests that we pick elements $i \in \mathbb{N}$ such that $p_{\in}(i) > 0.5$. It is important to note that, given $A = \{\langle p_{\in}(i), p_{\notin}(i), p_I(i)\rangle\}_{i \in \mathbb{N}}$, not every i will be enumerated if we use this probability criterion. However, changing the criterion depending on what aspect of the set we want to look at and depending on the application, would also change the enumerated set. Therefore, we would need a kind of criterion function to set a probability threshold regarding which elements of the neutrosophic set are to be enumerated.

In practice, one often encounters a situation where the given information is not directly used but rather analyzed under the criterion determined by a function. We examine how the computation behaves when we impose a function on the neutrosophic oracle tape. That is, suppose that $f : \mathbb{N} \to \{a, b, c\}$ is a function, where $a, b, c \in \mathbb{Q}$, which maps each cell of the neutrosophic oracle tape to a probability value. For example, f could be defined as a constant non-membership function which assigns every triplet in the cells to the non-membership $\notin$ attribute. In this case, the probability of any natural number not being an element of the considered oracle A is just 1. When these kinds of functions are used in the oracle information of A, we may be able to compute some useful information.

The intuition in using criterion functions is to select, under a previously determined probability threshold, a natural number from the probability distribution which is available in a given neutrosophic subset of natural numbers. As an example, let us imagine a neutrosophic subset A of natural numbers. Suppose for simplicity that A is finite and is defined as:

$$A = \{\langle 0.1, 0.4, 0.5\rangle, \langle 0.6, 0.3, 0.1\rangle, \langle 0, 0.9, 0.1\rangle\}.$$

First of all, we should read this as follows: A has neutrosophic information about the first three natural numbers 0, 1, 2. In this example, $p_{\in}(0) = 0.1$, $p_{\notin}(0) = 0.4$, $p_I(0) = 0.5$. For the natural number 1, we have that $p_{\in}(1) = 0.6$, $p_{\notin}(1) = 0.3$, $p_I(1) = 0.1$. Finally, for the natural number 2, we have $p_{\in}(2) = 0$, $p_{\notin}(2) = 0.9$, $p_I(2) = 0.1$. Now if we want to know which natural numbers are in A, normally we would only pick the number 2 since $p_{\in}(2) > 0.5$. Our criterion of enumeration in this case is 0.5. In general, this probability value may not be always applicable. Moreover, this probability threshold value may not be constant. That is, we may want to have a different probability threshold for every natural number i. If our criterion were to select the ith element whose probability exceeds p_i, we would enumerate those numbers. For example, if the criterion is defined as

$$f(0) = 0, \quad f(1) = 0.8, \quad f(2) = 0.2,$$

then for the first triple, the probability threshold is 0, meaning that we enumerate the natural number 0 if $p_\in(0) > 0$. Obviously 0 will be enumerated in this case since $p_\in(0) = 0.1 > 0$. The probability threshold for enumerating the number 1 is 0.8, so it will not be enumerated since $p_\in(1) = 0.6 < 0.8$. Finally, the probability threshold for enumerating the number 2 is 0.2. In this case, 2 will not be enumerated since $p_\in(2) = 0 < 0.2$. So the enumeration set for A under the criterion f will be $\{0\}$.

We are now ready to give the formal definition of a criterion function.

Definition 5. *A criteria function is a mapping $f : \mathbb{N} \to \mathbb{Q}$ which, given a neutrosophic subset A of natural numbers, determines a probability threshold for each triple $\langle p_\in(i), p_\notin(i), p_I(i)\rangle$ in A.*

We first note a simple observation that if the criterion function is the constant function $f(n) = 0$ for any $n \in \mathbb{N}$, the enumeration set will be equal to $\mathbb{N}$ itself. However, this does not mean that the enumeration set will be empty if $f(n) = 1$. Given a neutrosophic set A, if $p_\in(i) = 1$ for all i, then the enumeration set for A will also be equal to $\mathbb{N}$.

We shall next give the following theorem. First we remind the reader that we call a function f *strictly decreasing* if $f(i+1) < f(i)$.

Theorem 3. *Let f be a strictly decreasing criterion function for a neutrosophic set A such that $p_\in(i) < p_\in(i+1)$ for every $i \in \mathbb{N}$, and let $\mathcal{E}_A$ be the enumeration set for A under the criterion f. Then, there exists some $k \in \mathbb{N}$ such that $|\mathcal{E}_A| < k$.*

Proof. Clearly, given A and that for each i, $p_\in(i) < p_\in(i+1)$, only those numbers i which satisfy $p_\in(i) > f(i)$ will be enumerated. Since the probability distribution of membership degrees of elements of A strictly increases and f is strictly decreasing, there will be some number $j \in \mathbb{N}$ such that $p_\in(i) \leq f(j)$. Moreover, for the same reason $p_\in(m) \leq f(m)$ for every $m > j$. Therefore, the number of elements enumerated is less than j. That is, $|\mathcal{E}_A| < j$. □

We denote the complement of a neutrosophic subset of natural numbers A by A^c and we define it as follows. Let $p^A_\in(i)$ denote the probability of i being an element of A and let $p^A_\notin(i)$ denote the probability of i being not an element of A. In addition, $p^A_I(i)$ denotes the probability of the membership of i being indeterminate. Then:

$$A^c(i) = \langle p^A_\notin(i), p^A_\in(i), p^A_I(i)\rangle.$$

So the complement of a neutrosophic set in consideration is formed by simply interchanging the probabilities of membership and non-membership for all $i \in \mathbb{N}$. Notice that the probability of indeterminacy remains the same. Our next observation is as follows. Suppose that A and A^c are neutrosophic subsets of natural numbers and f is a criterion function. If $\mathcal{E}_A \subset \mathcal{E}_{A^c}$, then clearly $p_\notin(i) \geq p_\in(i)$ for all $i \in \mathbb{N}$.

The probability distribution of members of a neutrosophic set can be also be given by a function $g(i,j)$ such that $i \in \mathbb{N}$ and $j \in \{1,2,3\}$ where j is the index for denoting the *membership* probability by 1, *non-membership* probability by 2, and *indeterminacy* probability by 3, respectively. For instance, for $i \in \mathbb{N}$, $g(i,2)$ denotes the probability of the non-membership of i generated by the function g. Now g being a computable function means, for any i, j, there is an algorithm to find the value of $g(i,j)$. We give the following theorem.

Theorem 4. *Let A be a neutrosophic set. If g is a computable function, then there exists a computable criterion function f such that $\mathcal{E}_A$ is the enumeration set of A under the criterion function f and, moreover, $\mathcal{E}_A = \mathbb{N}$.*

Proof. Suppose that we are given A. If g is a computable function that generates the probability distribution of members of A, then we can computably find $g(i,1) = k$. We then simply let $f(i)$ be some $m \leq k$. Since $p_{\in}(i) = k \geq f(i)$, every i will be a member of $\mathcal{E}_A$. Since i is arbitrary, $\mathcal{E}_A = \mathbb{N}$. □

We say that a function f *majorizes* a function g if $f(x) > g(x)$ for all x. Suppose now that g is a quickly growing function in the sense that it majorizes every computable function. That is, assume that $g(i,j) \geq f(i)$ for every $i, j \in \mathbb{N}$ and every computable function f. Now in this case g is necessarily non-computable. Otherwise we would be able to construct a function h where $h(i)$ is chosen to be some $s > t$ such that $g(i,j)$ is defined at step t. So if g is not computable, we cannot apply the previous theorem on g. The only way to enumerate A is by using relative computability rather than giving a plain computable procedure. Suppose that we are given such a function g. Let $\Psi_i(A;i)$ denote the ith Turing machine with oracle A and input i. We define $g' = \{x : \Psi_x(g;x)\downarrow\}$ to be the *jump* of g, where $x = \langle i,j \rangle$ for a uniform pairing function $\mathbb{N} \times \mathbb{N} \to \mathbb{N}$. The jump of g is basically the halting set relativized to g. If we want to enumerate members of A, we can then use g' as an oracle. Since, by definition, g' computes g, we enumerate members of A computably *in* g'.

We shall also note an observation regarding the relationship between A and A^c. Given a function f, unless $f(i)$ is strictly between $p^A_{\in}(i)$ and $p^A_{\notin}(i)$, we have that $\mathcal{E}_A = \mathcal{E}_{A^c}$. That is, the only case when $\mathcal{E}_A \neq \mathcal{E}_{A^c}$ is if $p^A_{\in}(i) < f(i) < p^A_{\notin}(i)$ or $p^A_{\notin}(i) < f(i) < p^A_{\in}(i)$. Let us examine each case. In the first case, since $f(i) > p^A_{\in}(i)$, i will not be enumerated into $\mathcal{E}_A$, but since $p^{A^c}_{\in}(i) > f(i)$, it will be enumerated into $\mathcal{E}_{A^c}$. The second case is just the opposite. That is, i will be enumerated into $\mathcal{E}_A$ but not into $\mathcal{E}_{A^c}$.

What about the cases where i is enumerated into both enumeration sets? It depends on how we allow our criteria function to operate over probability distributions. If we only want to enumerate those elements i such that $p_{\in}(i) \geq f(i)$, then we may have equal probability distribution among membership and non-membership attributes. We may have that $p_{\in}(i) = p_{\notin}(i) = 0.5$ and $p_I(i) = 0$. In this case, we get to enumerate i both into $\mathcal{E}_A$ and $\mathcal{E}_{A^c}$. However, if we allow the criterion function to operate in a way that i is enumerated if and only if $p_{\in}(i) > f(i)$, then it must be the case that $p_{\notin}(i) < f(i)$ so i will only be enumerated into $\mathcal{E}_A$.

The use of the criterion function may vary depending on the application and which aspect of the given neutrosophic set we want to analyze.

4. Conclusions

We introduced the neutrosophic counterpart of oracle Turing machines with neutrosophic values allowed in the oracle tape. For this we presented a new type of oracle tape where each cell contains a triplet of three probability values, namely for the membership, non-membership, and indeterminacy. The notion of neutrosophic oracle Turing machine is interesting in its own right since oracle information is used in relative computability of sets and enables us to investigate the computability theoretic properties of sets relative to one another. In this paper, we also introduced a method to enumerate the elements of a neutrosophic subset of natural numbers. For this we defined a criterion function to choose elements which satisfy a certain probability degree. This defines a method that can be used in many applications of neutrosophic sets, particularly in decision making problems, solution space searching, and many more. We proved some results about the relationship between the enumeration sets of a given neutrosophic subset of natural numbers and the criterion function. A future work of this study is to investigate the properties of equivalence classes induced by the operator $\equiv_N$. We may call this equivalence class, neutrosophic degree of computability. It would be interesting to study the relationship between neutrosophic degrees of computability and classical Turing degrees. The results also arise further developments in achieving of new generation of computing machines such as fuzzy cellular nonlinear networks paradigm or the memristor-based cellular nonlinear networks [9]. The latter of course has practical benefits.

Author Contributions: Conceptualization, A.Ç. and S.T.; Methodology, A.Ç.; Validation, A.Ç., S.T., and F.S.; Investigation, A.Ç. and S.T.; Resources, A.Ç., S.T., and F.S.; Writing-original draft preparation, A.Ç and S.T.; Writing-review & editing, A.Ç., S.T., and F.S.; Supervision, A.Ç., S.T. and F.S.; Funding Acquisition, F.S.

Funding: This research received no external funding

Conflicts of Interest: The authors declare no conflict of interest

References

1. Turing, A.M. On Computable Numbers With an Application to the Entscheidungsproblem. *Proc. Lond. Math. Soc.* **1937**, *s2-42*, 230–265. [CrossRef]
2. Cooper, S.B. *Computability Theory*; Chapman & Hall, CRC Press: Boca Raton, FL, USA; New York, NY, USA; London, UK, 2004.
3. Downey R.; Hirshfeldt, D. *Algorithmic Randomness and Complexity*; Springer: Berlin, Germany, 2010.
4. Soare, R.I. *Recursively Enumerable Sets and Degrees*; Perspectives in Mathematical Logic; Springer: Berlin, Germany, 1987.
5. Gödel, K. Über Formal Unentscheidbare Sätze der Principia Mathematica und Verwandter Systeme I. *Monatshefte für Mathematik und Physik* **1931**, *38*, 173–198. [CrossRef]
6. Turing, A.M. Systems of logic based on ordinals. *Proc. Lond. Math. Soc.* **1939**, *45*, 161–228. [CrossRef]
7. Smarandache, F. *Neutrosophy. Neutrosophic Probability, Set, and Logic*; American Research Press: Rehoboth, DE, USA, 1998; pp. 104–106.
8. Smarandache, F. A generalization of the intuitionistic fuzzy set. *Int. J. Pure Appl. Math.* **2005**, *24*, 287–297.
9. Buscarino, A.; Corradino, C.; Fortuna, L.; Frasca, M.; Chua, L.O. Turing patterns in memristive cellular nonlinear networks. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2016**, *63*, 1222–1230. [CrossRef]