Fuzzifying P Systems

Apostolos Syropoulos

Greek Molecular Computing Group
366, 28th October Str.
GR-671 00 Xanthi, GREECE
apostolos@araneous.gr

Abstract. Uncertainty is an inherent property of all living systems. Curiously enough, computational models inspired by biological systems do not take, in general, under consideration this essential aspect of living systems. In this paper, after introducing the notion of a *multi-fuzzy* set (i.e., multisets where objects are repeated to some degree), we introduce two variants of P systems with fuzzy components: P systems with fuzzy data and P systems with fuzzy multiset rewriting rules. By silently assuming that fuzzy data are not the result of some fuzzification process, P systems with fuzzy data are shown to be a step towards real hypercomputation. On the other hand, P systems with fuzzy multiset rewriting rules are shown to be equivalent to fuzzy Turing machines. The paper concludes with remarks concerning the present work and future research.

In science we try to tell people things in such a way that they understand something that nobody knew before. In art, one takes something everyone knows and tries to express it in ways nobody ever thought before.
Based on a dialogue from *The One True Platonic Heaven: A Scintific Fiction on the Limits of Knowledge*, John L. Casti, Joseph Henry Press, Washington, D.C., 2003.

1 Introduction

P systems [11], a new promising model of computation, inspired by the way cells live and function, are built around the notion of nested compartments surrounded by porous *membranes* (hence the term *membrane computing*). It is quite instructive to think of the membrane structure as a "generalized" babushka dolls toy (see Figure 1), where each doll may contain more than one doll. Initially, each compartment contains a number of possible repeated objects (i.e., multisets of objects). Once "computation" commences, the compartments exchange objects according to a number of multiset processing rules that are associated with each compartment; in the simplest case, these processing rules are just multiset rewriting rules. The activity stops when no rule can be applied any more.

The result of the computation is equal to the number of objects that reside in a designated compartment called the *output membrane*. Equivalently, we can construct a grammar simulating a given system Π. Although there are many different forms of P systems, these will not concern us here. The interested readers should consult the standard reference on P systems [12] for more information.

Fig. 1. A babushka dolls toy can be used to provide an instructive description of the membrane structure of P systems.

Fuzzy set theory is a theory that generalizes the concept of the set (for an overview, see for example [6]). In fuzzy set theory, an element of a fuzzy subset belongs to it to a degree, which is usually a number between 0 and 1. Ever since its inception by Lotfi Asker Zadeh, fuzzy set theory prompted many researchers to *fuzzify* (i.e., to "soften" rigid categorization of) other mathematical structures. Practically, this meant that, during the fuzzification process of a particular structure, one had to "soften" the way the properties of a particular member of a group related to the properties of the group. Fortunately, the subsequent fuzzification "storm" was not just based on a mere mathematical curiosity, but on real grounds. For example, rigid mathematical models employed in biology are not completely adequate for the interpretation of biological information. This fact has led to the adoption of new fuzzy models and methodologies (for example, see [18]). Consequently, the fuzzification of P systems is a quite reasonable development. Indeed, Păun in [12, page 365] discusses the idea of *approximate* computing, in the framework of the theory of P systems (i.e., P systems with non-classical components). Now, if we are interested in fuzzifying P systems, we need to fuzzify one or all of their characteristics. In particular, this means that we can have P systems with fuzzy processing rules and/or with fuzzy data. However, since the theory of P systems does not make any assumption regarding the size of the compartments, it makes no sense to fuzzify the notion of the membrane.

In his seminal paper from 1986 [19], Yager introduced *fuzzy multisets*, that is a multiset of pairs (x, i), each of them denoting the degree i to which x belongs to some universe X (see [14] for an overview of the theory of multisets). Clearly, this mathematical structure is a multiset in $X \times L$, where L is a *frame*. An algebra $(L; \vee, \wedge)$ is called a frame if

- L is a nonempty set;
- $\wedge$ and $\vee$ are binary operations on L;
- both $\wedge$ and $\vee$ are
  - idempotent (i.e., $x \vee x = x$ and $x \wedge x = x$);
  - commutative (i.e.,$x \wedge y = y \wedge x$ and $x \vee y = y \vee x$);
  - associative (i.e., $(x \wedge y) \wedge z = x \wedge (y \wedge z)$ and $(x \vee y) \vee z = x \vee (y \vee z)$);
  - distributive (i.e., $(x \wedge y) \vee (x \wedge z) = x \wedge (y \vee z)$ and $(x \vee y) \wedge (x \vee z) = x \vee (y \wedge z)$);
  - and they satisfy the absorption law (i.e., $x \wedge (x \vee y) = x \vee (x \wedge y) = x$).

However, one can go the other way and define a mathematical structure where each element occurs a number of times to some degree. We call these structures *multi-fuzzy sets*. From a mathematical point of view the two structures are very similar, but from a conceptual point of view they are completely different structures—in the former case, elements having a particular property to the same degree may occur a number of times, while in the later case, identical elements may occur a number of times up to some truth degree.

In section 4 we present P systems with fuzzy data that are capable of computing real numbers. Naturally, the idea of computational devices that compute real numbers is not novel. For example, Alan Turing studied real-number computability since 1936. Also, Blum, Shub and Smale [1] have developed the so-called BSS-machine (i.e., a sort of Turing machine) that is capable of handling real numbers and real number functions. A common characteristic of these computational models is they transcend the capabilities of Turing machines. In addition, we should note that Wegner in [16] discusses *super-Turing* computation in the framework of interactive computation, while Kieu [4] advertises the idea that quantum computers are able to solve problems which cannot be computed by the universal Turing machine. This "peculiarity" of the BSS-machines and the related computational models has led a number of researchers and thinkers to manifesto that *hypercomputation* [2] includes models of computation that are indeed more powerful than Turing machines. Consequently, they claim that hypercomputation falsifies the Turing-Church thesis. However, we should note that not everybody shares this idea. For example, Cotogno [3] believes that real-number computation does not really go beyond the Church-Turing barrier.

Structure of the paper. We start by defining multi-fuzzy sets, possible extensions of the concept and their properties. In addition, we present the theory of fuzzy grammars and the concept of a fuzzy rewriting rule. Next, P systems with fuzzy data (i.e., P systems where each compartment is populated with multi-fuzzy sets) are defined. We continue with the presentation of P systems with fuzzy multiset rewriting rules. Furthermore, there is a brief discussion about P systems with both fuzzy data and fuzzy rewriting rules. We conlcude with remarks concerning the present work and future research.

2 Fuzzifying Multisets

As it has been noted in the introduction, fuzzy multisets have been introduced by Yager. Formally, a fuzzy multiset is a multiset of pairs (x, i), where $x \in X$, $i \in [0, 1]$ and X is an arbitrary set. The pair (x, i) denotes that the element x belongs to the set X with degree equal to i. More generally, one can replace the unit interval with a frame L and get L-fuzzy-multisets. Note that a partially ordered set is a frame iff

(i) every subset has a join
(ii) every finite subset has a meet
(iii) binary meets distribute over joins:

$$x \wedge \bigvee Y = \bigvee \{x \wedge y : y \in Y\}.$$

Miyamoto has exemplified in his work (see [8]) that Yager's definitions are somehow inadequate, since one cannot easily perform the basic multiset operations (e.g., intersection, union, etc.). Thus, he proposed a better formulation, which, however, does not change the essence of the initial definition.

Intuitively, fuzzy multisets model the case where otherwise indistinguishable objects possess a particular property to a certain degree. If we go the other way and fuzzify the number of occurrences of each object, then we get a new structure, which we call a *multi-fuzzy set*. The motivation for introducing these structures was quite practical. We wanted to fuzzify the basic property of multisets and, thus, to define multisets where the number of occurances form a fuzzy set. Let us now proceed with the formal definition:

Definition 1. *Suppose that X is a (fixed) universe, then a multi-fuzzy set is a function $\mathcal{A} : X \to \mathbb{N}_0 \times \mathrm{I}$, where $\mathbb{N}_0$ is the set of all positive integers including zero and I is the unit interval $[0, 1]$. The expression $\mathcal{A}(x) = (n, i)$ denotes that the degree to which x occurs n times in the multi-fuzzy set is equal to i.*

Obviously, one can go further and extend the definition above. For example, in [15] we introduce multi-fuzzy sets where both the membership and the non-membership degrees are considered. However, it is not yet clear how this may affect the theory developed in the rest of this paper.

Starting from a multi-fuzzy set $\mathcal{A}$, we can define the following two functions: the *multiplicity* function $\mathcal{A}_m : X \to \mathbb{N}_0$ and the *membership* function $\mathcal{A}_\mu : X \to I$. Obviously, if $\mathcal{A}(x) = (n, i)$, then $\mathcal{A}_m(x) = n$ and $\mathcal{A}_\mu(x) = i$.

Remark 1. Any ordinary set A is identical to the multi-fuzzy set $\mathcal{A}$ defined as follows:

$$\mathcal{A}(a) = (\chi_A(a), 1), \forall a \in X,$$

where χ_A is the characteristic function of A. In addition, any fuzzy set $A : X \to \mathrm{I}$ is identical to the multi-fuzzy set $\mathcal{A}'$ defined as follows:

$$\mathcal{A}'(a) = (1, A(a)), \forall a \in X.$$

Similarly, any multiset $M : X \to \mathbb{N}_0$ can be represented by the multi-fuzzy set:

$$\mathcal{M}(a) = (M(a), 1), \forall a \in X.$$

The cardinality of multi-fuzzy sets is defined as follows:

Definition 2. *Suppose that $\mathcal{A}$ is a multi-fuzzy set having the set X as its universe, then its cardinality, denoted* $\operatorname{card}\mathcal{A}$*, is defined as*

$$\operatorname{card}\mathcal{A} = \sum_{a \in A} \mathcal{A}_m(a)\mathcal{A}_\mu(a).$$

Remark 2. Obviously, the previous definition gives the expected results for the special cases we discussed in the remark above.

Operations on multi-fuzzy sets. Assume that $\mathcal{X}$ and $\mathcal{Y}$ are two multi-fuzzy sets with universe the set Z, then their union, intersection, sum and their difference are defined as follows:

Definition 3. ***(Union of multi-fuzzy sets)***

$$\big(\mathcal{X} \cup \mathcal{Y}\big)(z) = \Big(\max\Big\{\mathcal{X}_m(z), \mathcal{Y}_m(z)\Big\}, \max\Big\{\mathcal{X}_\mu(z), \mathcal{Y}_\mu(z)\Big\}\Big).$$

Definition 4. ***(Intersection of multi-fuzzy sets)***

$$\big(\mathcal{X} \cap \mathcal{Y}\big)(z) = \Big(\min\Big\{\mathcal{X}_m(z), \mathcal{Y}_m(z)\Big\}, \min\Big\{\mathcal{X}_\mu(z), \mathcal{Y}_\mu(z)\Big\}\Big).$$

Definition 5. ***(Sum of multi-fuzzy sets)***

$$\big(\mathcal{X} \uplus \mathcal{Y}\big)(z) = \Big(\mathcal{X}_m(z) + \mathcal{Y}_m(z), (\mathcal{X}_\mu + \mathcal{Y}_\mu)(z)\Big).$$

where $(\mathcal{X}_\mu + \mathcal{Y}_\mu)(z) = (\mathcal{X}_\mu(z) + \mathcal{Y}_\mu(z)) - \mathcal{X}_\mu(z) \cdot \mathcal{Y}_\mu(z)$.

Definition 6. ***(Difference of multi-fuzzy sets)***

$$\big(\mathcal{X} \ominus \mathcal{Y}\big)(z) = \Big(\max\Big\{\mathcal{X}_m(z) - \mathcal{Y}_m(z), 0\Big\}, \min\Big\{\mathcal{X}_\mu(z), \mathcal{Y}_\mu(z)\Big\}\Big).$$

Some properties of the operations between multi-fuzzy sets are presented below:

Theorem 1. *For any three multi-fuzzy sets $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ having Z as their common universe, the following equalities hold:*

(i) Commutativity:

$$\mathcal{A} \cup \mathcal{B} = \mathcal{B} \cup \mathcal{A}$$
$$\mathcal{A} \cap \mathcal{B} = \mathcal{B} \cap \mathcal{A}$$
$$\mathcal{A} \uplus \mathcal{B} = \mathcal{B} \uplus \mathcal{A};$$

(ii) Associativity:

$$\mathcal{A} \cup (\mathcal{B} \cup \mathcal{C}) = (\mathcal{A} \cup \mathcal{B}) \cup \mathcal{C}$$
$$\mathcal{A} \cap (\mathcal{B} \cap \mathcal{C}) = (\mathcal{A} \cap \mathcal{B}) \cap \mathcal{C}$$
$$\mathcal{A} \uplus (\mathcal{B} \uplus \mathcal{C}) = (\mathcal{A} \uplus \mathcal{B}) \uplus \mathcal{C};$$

(iii) Idempotency:

$$\mathcal{A} \cup \mathcal{A} = \mathcal{A}$$
$$\mathcal{A} \cap \mathcal{A} = \mathcal{A};$$

(iv) Distributivity of union and intersection:

$$\mathcal{A} \cup (\mathcal{B} \cap \mathcal{C}) = (\mathcal{A} \cup \mathcal{B}) \cap (\mathcal{A} \cup \mathcal{C})$$
$$\mathcal{A} \cap (\mathcal{B} \cup \mathcal{C}) = (\mathcal{A} \cap \mathcal{B}) \cup (\mathcal{A} \cap \mathcal{C});$$

(v) Distributivity of sum:

$$\mathcal{A} \uplus (\mathcal{B} \cup \mathcal{C}) = (\mathcal{A} \uplus \mathcal{B}) \cup (\mathcal{A} \uplus \mathcal{C})$$
$$\mathcal{A} \uplus (\mathcal{B} \cap \mathcal{C}) = (\mathcal{A} \uplus \mathcal{B}) \cap (\mathcal{A} \uplus \mathcal{C});$$

Proof. We will prove only that the operators are commutative, as the other properties can be proved similarly.

$$\begin{aligned}
\big(\mathcal{A} \cup \mathcal{B}\big)(z) &= \Big(\max\Big\{\mathcal{A}_m(z), \mathcal{B}_m(z)\Big\}, \max\Big\{\mathcal{A}_\mu(z), \mathcal{B}_\mu(z)\Big\}\Big)\\
&= \Big(\max\Big\{\mathcal{B}_m(z), \mathcal{A}_m(z)\Big\}, \max\Big\{\mathcal{B}_\mu(z), \mathcal{A}_\mu(z)\Big\}\Big)\\
&= \big(\mathcal{B} \cup \mathcal{A}\big)(z)\\
\big(\mathcal{A} \cap \mathcal{B}\big)(z) &= \Big(\min\Big\{\mathcal{A}_m(z), \mathcal{B}_m(z)\Big\}, \min\Big\{\mathcal{A}_\mu(z), \mathcal{B}_\mu(z)\Big\}\Big)\\
&= \Big(\min\Big\{\mathcal{B}_m(z), \mathcal{A}_m(z)\Big\}, \min\Big\{\mathcal{B}_\mu(z), \mathcal{A}_\mu(z)\Big\}\Big)\\
&= \big(\mathcal{B} \cap \mathcal{A}\big)(z)\\
\big(\mathcal{A} \uplus \mathcal{B}\big)(z) &= \Big(\mathcal{A}_m(z) + \mathcal{B}_m(z), \Big(\mathcal{A}_\mu + \mathcal{B}_\mu\Big)(z)\Big)\\
&= \Big(\mathcal{B}_m(z) + \mathcal{A}_m(z), \Big(\mathcal{B}_\mu + \mathcal{A}_\mu\Big)(z)\Big)\\
&= \big(\mathcal{B} \uplus \mathcal{A}\big)(z)
\end{aligned}$$

Clearly, the equality $(\mathcal{A}_\mu + \mathcal{B}_\mu)(z) = \Big(\mathcal{B}_\mu + \mathcal{A}_\mu\Big)(z)$ follows from the commutativity of addition and multiplication. □

Going one step ahead. Let us now see how we can define a multi-fuzzy set whose elements may occur a number of different times to some degree. Here is a formal definition:

Definition 7. *The fuzzy relation $\mathcal{X} : X \times \mathbb{N}_0 \to \mathrm{I}$ is a **general multi-fuzzy set**. In particular, the expression $\mathcal{X}(x, n) = i$ denotes that the degree to which x occurs n times in the multi-fuzzy set is equal to i.*

Of course, one can extend the previous definition and introduce general L-multi-fuzzy sets, but since the definition is straightforward, we omit it mainly for reasons of brevity.

3 Fuzzy Rewriting Rules

If we consider the multisets contained in the various compartments of a P system as the data of a computer program, then the evolution rules are the instructions that make up the program. We have already managed to provide ways to fuzzify the data of our programs. Now we will see how we can fuzzify the instructions.

The obvious way to fuzzify the multiset rewriting rules associated with each compartment is to assign an "execution" degree to each rule. This execution degree will denote the degree to which a given applicable rule can be used in a particular step. At first glance this construction may seem familiar. One may take a P system with fuzzy multiset rewriting rules for a probabilistic P system (see [10] for an overview and [7] for a first study of probabilistic rewriting P systems). However, this assumption is completely false. First of all, in a probabilistic system the minimum requirement is that the probabilities of all rules must add together to one, while this is not necessary for the "execution" degrees assigned to each rule. More generally, fuzzy set theory deals with the likelihood of an event, while probability theory with the extend of that event. In fact, from a mathematical perspective, fuzzy sets and probability exist as parts of a greater Generalized Information Theory that includes many formalisms for representing uncertainty (including random sets, Demster-Shafer evidence theory, probability intervals, possibility theory, general fuzzy measures, interval analysis, etc.). Furthermore, one can also talk about random fuzzy events and fuzzy random events. This whole issue is beyond the scope of this paper, and the reader should refer to [5] for details.

In order to understand fuzzy rewriting rules, we need to grasp the notion of a fuzzy grammar. The following definition is a more formal version of the corresponding defintion found in [6]:

Definition 8. *A fuzzy grammar,* FG*, is defined by the quintuple*

$$\mathrm{FG} = (V_N, V_T, S, P, A)$$

where

- *V_N is the set of nonterminal symbols;*
- *V_T is the set of terminal symbols ($V_T \cap V_N = \emptyset$);*
- *$S \in V_N$ is the starting symbol;*
- *P is a finite set of production rules of the form $\alpha \to \beta$, where $\alpha \in (V_T \cup V_N)^* V_N (V_T \cup V_N)^*$ and $\beta \in (V_T \cup V_N)^*$ (i.e., α must contain at least one symbol from V_N); and*
- *A is a fuzzy subset*
$$A : P \to \mathrm{I}$$
The value $A(p)$ is the grade of applying a production $p \in P$.

For $\sigma, \psi \in (V_T \cup V_N)^*$, σ is said to be a *direct derivative* of ψ, written as $\psi \Rightarrow \sigma$, if there are (possibly empty) strings ϕ_1 and ϕ_2 such that $\psi = \phi_1 \alpha \phi_2$, $\sigma = \phi_1 \beta \phi_2$, and $\alpha \to \beta$ is a production of the grammar. The string ψ produces σ, written as $\psi \overset{+}{\Rightarrow} \sigma$ if there are strings $\phi_0, \phi_1, \ldots, \phi_n$ $(n > 0)$, such that

$$\psi = \phi_0 \Rightarrow \phi_1, \phi_1 \Rightarrow \phi_2, \ldots, \phi_{n-1} \Rightarrow \phi_n = \sigma$$

A string $\alpha \in V_T^*$ is a *sentential form* of FG if it is a derivative of the unique nonterminal symbol S.

A string $\alpha \in V_T^*$ is said to belong to the fuzzy language $L(\mathrm{FG})$ if and only if α is a sentential form. In addition, the degree to which α belongs to $L(\mathrm{FG})$ is

$$\max_{1 \le k \le n} \min_{1 \le i \le \ell_k} A(p_i^k), \tag{1}$$

where n is the number of different derivatives, ℓ_k is the length of the kth derivative, and p_i^k denotes the ith direct derivative in the kth derivative ($i = 1, 2, \ldots, \ell_k$).

It is not difficult to extend the previous definition. So instead of the standard intersection and union operators (i.e., min and max, respectively), one can use a t-norm and the corresponding t-conorm. A t-norm $\sqcap : \mathrm{I} \times \mathrm{I} \to \mathrm{I}$ is a function with the following properties:

(i) $a \sqcap 1 = a$,
(ii) $b \le c$ implies $a \sqcap b \le a \sqcap c$,
(iii) $a \sqcap b = b \sqcap a$,
(iv) $a \sqcap (b \sqcap c) = (a \sqcap b) \sqcap c$,

while a t-conorm $\sqcup : \mathrm{I} \times \mathrm{I} \to \mathrm{I}$ is a function with the following properties:

(i) $a \sqcup 0 = a$,
(ii) $b \le c$ implies $a \sqcup b \le a \sqcup c$,
(iii) $a \sqcup b = b \sqcup a$,
(iv) $a \sqcup (b \sqcup c) = (a \sqcup b) \sqcup c$.

Now, equation 1 can be rewritten more generally as

$$\bigsqcup_{k=1}^{n} \bigsqcap_{i=1}^{\ell_k} A(p_i^k).$$

Since rewriting rules are actually productions that are applied to a starting string repeatedly, a fuzzy rewriting rule is just a crisp rewriting rule associated with a truth degree:

Definition 9. *Let V be a finite set of symbols, then a fuzzy rewriting rule has the following form*

$$\alpha \xrightarrow{\rho} \beta,$$

where $\alpha, \beta \in V^*$ and $\rho \in \mathrm{I}$ indicates the plausibility that α is reduced to β in a derivation step.

4 P Systems With Fuzzy Data

In the previous sections it was shown how one can fuzzify the basic "ingredients" of P systems, that is the multisets and the multiset rewriting rules. In this section we show how we can build P systems with fuzzy data. But first, let us explain why we have opted to use multi-fuzzy sets instead of fuzzy multisets.

It is a fact that the number computed by a P systems is equal to the cardinality of the multiset contained in the output membrane. Ergo, a P system with fuzzy data is one where the various objects occur to some degree. Certainly, one may argue that using fuzzy multisets is a more natural choice. Our response to such an argument is that in a physical environment in most cases we are sure about the nature of objects, but we are not sure about the number of objects. In addition, if we were to adopt fuzzy multisets, then we would have P systems with *structured* objects. That is, we would have P systems where each object would be a pair (x, i), which, in turn, can be viewed as a new object x_i. Let us now see what is the effect of applying (crisp) multiset rewriting rules to fuzzy data.

Suppose that we have a membrane structure and each compartment is populated with a multi-fuzzy set. In addition, assume that each compartment is associated with a finite number of multiset rewriting rules. Assume that the degree to which n copies of α exist in a designated compartment A is i; also the degree to which m copies of α exist in a designated compartment B is j. If there is a rule that moves α's from A to B, then, after using this rule, the degree to which the compartment B will contain a multi-fuzzy set with $n + m$ copies of α's will be equal to $i + j - ij$ (i.e., we sum up the two multi-fuzzy sets). In the end, the result of the computation is equal to the cardinality of the output compartment. However, here we are facing a very serious problem: the cardinality of the output membrane is usually a (positive) real number and as such it makes no sense (at least in the discrete case). One solution is to *defuzzify* the result by introducing a threshold parameter, $\lambda \in \mathrm{I}$, which can be used to define a crisp cardinality for the multi-fuzzy sets as follows:

$$\operatorname{card}_\lambda \mathcal{A} = \sum_{a \in A} d(\lambda, a)\mathcal{A}_m(a),$$

where $d(\lambda, a)$ is a *defuzzification* function

$$d(\lambda, a) = \begin{cases} 1, \text{if } \mathcal{A}_\mu(a) \geq \lambda \\ 0, \text{otherwise} \end{cases}$$

Equipped with the above definitions and remarks, we are ready to provide a formal definition of P systems with fuzzy data (the reader is assumed to be familiar with basic elements of membrane computing, for instance, from [12]):

Definition 10. *A P system with fuzzy data is a construction*

$$\Pi_{\mathrm{FD}} = (O, \mu, w^{(1)}, \ldots, w^{(m)}, R_1, \ldots, R_m, i_0, \lambda)$$

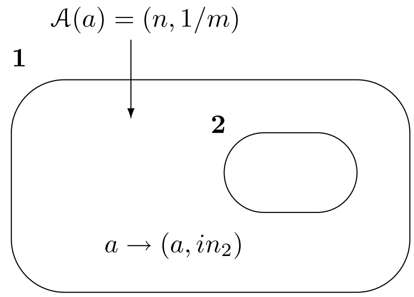
where:

(i) *O is an alphabet (i.e., a set of distinct entities) whose elements are called objects;*
(ii) *μ is the membrane structure of degree $m \geq 1$; membranes are injectivelly labeled with succeeding natural numbers starting with one;*
(iii) *$w^{(i)} : O \rightarrow \mathbb{N}_0 \times \mathrm{I}$, $1 \leq i \leq m$, are functions that represent multi-fuzzy sets over O associated with each region i;*
(iv) *R_i, $1 \leq i \leq m$, are finite sets of multiset rewriting rules (called evolution rules) over O. An evolution rule is of the form $u \rightarrow v$, $u \in O^*$ and $v \in O^*_{\mathrm{TAR}}$, where $O_{\mathrm{TAR}} = O \times \mathrm{TAR}$, $\mathrm{TAR} = \{\mathrm{here}, \mathrm{out}\} \cup \{\mathrm{in}_j | 1 \leq j \leq m\}$. The effect of each rule is the removal of the elements of the left-hand side of each rule from the "current" compartment and the introduction of the elements of right-hand side to the designated compartments;*
(v) *$i_0 \in \{1, 2, \ldots, m\}$ is the label of an elementary membrane (i.e., a membrane that does not contain any other membrane), called the output membrane; and*
(vi) *$\lambda \in [0, 1]$ is a threshold parameter, which is used in the final estimation of the computational result.*

Let us denote with $s_0^\Pi, s_1^\Pi, \ldots, s_n^\Pi$ the sequence of numbers computed by a P system Π with fuzzy/crisp data, then the following is a direct consequence of the definition of the cardinality of multi-fuzzy sets:

Proposition 1. *Assume that Π_{FD} is a P system with fuzzy data whose threshold parameter is λ. In addition, assume that Π is the corresponding P system with crisp data, then $s_i^{\Pi_{\mathrm{FD}}} \leq s_i^\Pi$ for all $i \in \mathbb{N}$.*

Although definition 10 is reasonable enough, it is not really clear whether it is necessary to defuzzify the result of the computation. Naturally, it makes sense to go on with the defuzzification process, once we have data that are the result of a fuzzification process. However, it is quite possible that the data are not the result of some fuzzification process. Thus, P systems with fuzzy data produce, in general, real positive numbers and so, unexpectedly, extend computability. Naturally, one may regard such systems as a form of hypercomputational machines. However, it is soon to jump into any definitive conlusions. As an example, let us consider the following P system with fuzzy data:



This P system contains n objects in compartment 1, which will be transferred into compartment 2. If we decide to skip the defuzzification step, the result of the computation (i.e., the cardinality of the multi-fuzzy set contained in compartment 2) is equal to n/m. Thus, the result of this particular computation is a positive rational number.

We have demonstrated that P systems with fuzzy data are capable of computing real numbers. Also, in the introduction we have discussed some foundational approaches to real number computation. Consequently, one may jump into the conclusion that P systems with fuzzy data are indeed a form of hypercomputation. However, we repeat that it is too early for such a definitive conclusion. For example, one may say that these systems have nothing to do with Nature. However, our responce to such a thesis is that P systems with fuzzy data are as real as P systems with crisp data.

5 P Systems With Fuzzy Multiset Rewriting Rules

The idea behind P systems with fuzzy multiset rewriting rules is the fuzzification of the macro-step process. In other words, by fuzzifying the multiset rewriting rules, we introduce a truth degree associated with each step. In the end, these degrees are used to estimate the truth degree of the computation.

A P system with fuzzy multiset rewriting rules and crisp data is just an ordinary P system that has, in addition, a corresponding fuzzy set for each set R_i of multiset rewriting. A P system with multiset fuzzy rewriting rules will compute a number to some degree. Clearly, such systems must also obey the so called *maximal parallelism* principle, that is the rules should be selected in such a way that only "optimal" output will be yielded. Thus, P systems with fuzzy data differ fundamentally from P systems with probabilist rewriting rules in that there is no bias in the selection of the rules.

Definition 11. *A P system with fuzzy multiset rewriting rules is a construct*

$$\Pi_{\mathrm{FR}} = (O, \mu, w_1, \ldots, w_m, R_1, \ldots, R_m, \sigma_1, \ldots, \sigma_m, i_0, \sqcap, \sqcup)$$

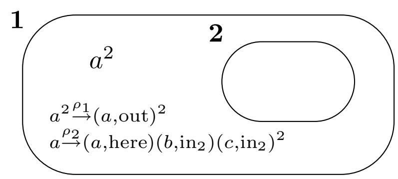
where each $\sigma_i : R_i \to \mathrm{I}$ is a fuzzy set defined over R_i, $\sqcap$ is a t-norm and $\sqcup$ a t-conorm.[1] *All other components are identical to those of an ordinary P system.*

[1] We assume here that both the t-norm and the t-conorm are *computable* in order to avoid problems of an entirely different nature. When we say computable we mean

When a P system with fuzzy multiset rewriting rules halts, the result of the computation up to some degree is equal to the cardinality of the multiset contained in the output compartment. Clearly, it is also necessary to know how to compute the truth degree that is associated with the computational result. More specifically:

Definition 12. *Assume that in the end of the computation the output compartment* i_0 *contains copies of the objects* β_1, $\beta_2,\ldots,$ β_n. *For each* β_i *we compute the quantity* $\rho_i^{\sqcup} = \sqcup_{j=1}^{k}\rho_j$, *where* ρ_j *is the "likelihood" degree of each of the rules* r_j *that produce* β_i. *The "likelihood" degree of the computation,* $\rho^{\sqcap}$, *is equal to* $\sqcap_{j=1}^{n}\rho_j^{\sqcup}$.

Example 1. Consider the following P system with fuzzy multiset rewrting rules:



Suppose that this system halts after n steps, then the crisp result of the computation will be equal to $6n$. Now, the degree to which $6n$ is the result of the fuzzy P system is just ρ_2. Note that this example is actually a fuzzy equivalent of the example given on [12, page 56].

Fuzzy Turing machines (for example, see [9, 17]) are computational models where each transition is associated with a truth degree. Clearly, it is interesting to see whether there is some connection between fuzzy Turing machines and P systems with fuzzy data. Although the notion of fuzzy Turing machines appeared a long time ago, still it is a concept that is not widely known.

Suppose that U and V are two non-empty sets and that $f : U \to V$ is a function, which is not necessarily total, then a W-function, f_W, associated to f is a (partial) function that maps elements from $U \times V$ to members of the semiring[2] $(W, \sqcap, \sqcup)$. More specifically, if $f(u) = v$, then $f_W(u, v)$ denotes the degree to which we are certain that the computation $f(u)$ yields the result v. Let us now proceed with the definition of the Santos type fuzzy Turing machine [13]:

---

that there is an effective procedure by means of it we can compute their values for all possible arguments.

[2] A semiring is a set together with two binary operators $(S, \oplus, \otimes)$ satisfying the following conditions:

(i) Additive associativity: For all $a, b, c \in S$, $(a\oplus) \oplus c = a \oplus (b \oplus c)$,
(ii) Additive commutativity: For all $a, b \in S$, $a \oplus b = b \oplus a$,
(iii) Multiplicative associativity: For all $a, b, c \in S$, $(a \otimes b) \otimes c = a \otimes (b \otimes c)$,
(iv) Left and right distributivity: For all $a, b, c \in S$, $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$ and $(b \oplus c) \otimes a = (b \otimes a) \oplus (c \otimes a)$.

Definition 13. *A Santos type fuzzy Turing machine is a septuple*

$$(S, Q, q_i, q_f, \delta, W, \delta_W).$$

where:

(i) S *represents a finite non-empty set of input symbols,*
(ii) Q *denotes a finite non-empty set of states such that* $S \cap Q = \emptyset$*,*
(iii) $q_i, q_f \in Q$ *are the symbols designating the initial and final state, respectively,*
(iv) $\delta \subset (Q \times S) \times (Q \times (S \times \{-1, 0, 1\}))$ *is the next-move relation,*
(v) W *is the semiring* $(W, \wedge, \vee)$*,*
(vi) $\delta_W : (Q \times S) \times (Q \times (S \times \{-1, 0, 1\})) \to W$ *is a W-function that assigns a degree of certainty to each machine transition.*

Assume that $\eta_W(C_i, C_{i+1})$ denotes the degree of reachability of C_{i+1} from C_i, then, in the case of a deterministic fuzzy Turing machine, the degree of certainty of a particular computation that starts from C_0 and finishes at C_n (denoted by $\Gamma(C_0, C_n)$) is given by the following formula:

$$\Gamma(C_0, C_n) = \eta_W(C_0, C_1) \wedge \eta_W(C_1, C_2) \wedge \cdots \wedge \eta_W(C_{n-1}, C_n).$$

In the case of a non-deterministic fuzzy Turing machine, $G(0, n)$ denotes the set of truth degrees of a computation that starts from C_0 and finishes at C_n. In addition, the truth degree of this computation is

$$\Gamma(C_0, C_n) = \bigvee_{\gamma \in G(0,n)}^{*} \gamma,$$

where $\bigvee^{*}$ denotes the transitive closure of $\vee$ (i.e., the smallest fuzzy relation that contains $\vee$ and is transitive).

Theorem 2. *For every P system with fuzzy data there is fuzzy Turing machine that computes exactly the same set of numbers.*

Sketch of Proof. It has been proved that fuzzy Turing machines and crisp Turing machine have exactly the same computational power. In addition, it is known that a class of P systems with multiset rewriting rules have *at least* the computational power of Turinng machines In particular, this class of P systems include system transition P systems and P systems with cooperating rules, systems with bi-stable catalysts, systems with plain or bi-stable catalysts and priorities among rules, systems with plain catalysts, permeability control and dissolution agents, and systems with non-cooperating rules that create rules during the computation. Clearly, both transition P system with fuzzy rewriting rules and their crisp counterparts produce the same output. They differ in that the former produce a computational result up to some truth degree, while the later produce the same result with a truth degree that is equal to one. From these remarks it is not difficult to see that the theorem holds true. □

Let us fuzzify everything! Depending on how we interpret P systems with fuzzy data, a P system with both fuzzy multiset rewriting rules and fuzzy data can be viewed as a computational device that halts to a certain degree and, in addition, computes a particular integer to some degree. On the other hand, if we assume that the outcome of the computation is a real number, then such systems just halt to a certain degree. We are not sure whether P systems with both fuzzy data and fuzzy multiset rewriting rules are really interesting as models of computation, but we believe that they should be of use in the modeling of living organisms.

6 Conclusions And Future Research Directions

Uncertainty is an inherent property of all living systems. P systems are models of computation inspired by the way the cell lives and functions. Thus, it is more than necessary to introduce uncertainty in models of computation that are based on biological systems. In this paper we have reported the results of our endeavor to fuzzify P systems (i.e., to introduce uncertainty in a well-established model of natural computation). In particular, we developed the theory of multi-fuzzy sets and presented the notion of a fuzzy multiset rewriting rule in order to define P systems with fuzzy components (i.e., fuzzy data and/or fuzzy multiset rewriting rules). In addition, we remarked taht if one skips the defuzzification process, which is not really necessary in all cases, the resulting P systems with fuzzy data are capable of computing real numbers, in general. Thus, P systems enter the realm of hypercomputation in an unexpected way. Also, it has been shown that P systems with fuzzy multiset rewriting rules are equivalent to fuzzy Turing machines. Furthermore, the idea of P systems with both fuzzy data and fuzzy multiset rewriting rules was briefly discussed.

Hypercomputation is about ways to refute the Church-Turing thesis by constructing new models of computation that can solve classically unsolvable problems. The fact that P systems with fuzzy data can be used to compute real-numbers is definitely not an indication that these systems refute the Church-Turing thesis. However, they provide a solid ground for further developing the theory in order to see what are the deeper implications of these new definitions. On the other hand, the fact that fuzzy computability is practically equivalent to crisp computability is yet another reason why P systems with fuzzy data deserve a deeper study.

Acknowledgments

The author acknowledges partial financial support via the ΠPENEΔ program of the Research Committee of the Democritus University of Thrace. In addition, I thank Paolo Cotogno, Gheorghe Păun and Krassimir Atanassov for their helpful comments and suggestions. Also, I thank the anonymous referee for his/her many comments and suggestions that have helped me to substantially improve the final version of this paper.

References

1. Blum, L., Cucker, F., Shub, M., and Smale, S. *Complexity and Real Computation*. Springer-Verlag, 1997.
2. Copeland, B. J. Hypercomputation. *Minds and Machines 12* (2002), 461–502.
3. Cotogno, P. Hypercomputation and the Physical Church-Turing Thesis. *Brit. J. Philosophy of Science 54* (2003), 181–223.
4. Kieu, T. D. Quantum Hypercomputation. *Minds and Machines 12* (2002), 541–561.
5. Klir, G., and Folger, T. *Fuzzy Sets, Uncertainty, and Information*. Prentice Hall, Englewood Cliffs, NJ, USA, 1987.
6. Klir, G. J., and Yuan, B. *Fuzzy Sets and Fuzzy Logic : Theory and Applications*. Prentice Hall (Sd), 1995.
7. Madhu, M. Probabilistic Rewriting P Systems. *International Journal of Foundations of Computer Science 14*, 1 (2003), 157–166.
8. Miyamoto, S. Fuzzy multisets and application to rough approximation of fuzzy sets. In *Proceedings of the Fourth International Workshop on Rough Sets, Fuzzy Sets, and Machine Discovery (RSFD'96)* (1996), pp. 255–260.
9. Moraga, C. Towards a Fuzzy Computability? *Mathware & Soft Computing 6* (1999), 163–172.
10. Obtułowicz, A., and Păun, G. (In search of) probabilistic P systems. *BioSystems 70*, 2 (2003), 107–121.
11. Păun, G. Computing with Membranes. *Journal of Computer and System Sciences 61*, 1 (2000), 108–143.
12. Păun, G. *Membrane Computing: An Introduction*. Springer-Verlag, Berlin, 2002.
13. Santos, E. S. Fuzzy Algorithms. *Information and Control 17*, 4 (1970), 326–339.
14. Syropoulos, A. Mathematics of Multisets. In *Multiset Processing*, C. Calude, G. Păun, G. Rozenberg, and A. Salomaa, Eds., no. 2235 in Lecture Notes in Computer Science. Springer-Verlag, Berlin, 2001, pp. 347–358.
15. Syropoulos, A., and Papadopoulos, B. K. Remarks on Non-Symmetric Multi-Fuzzy Sets. Submitted for publication, 2004.
16. Wegner, P. Interactive foundations of computation. *Theoretical Computer Science 192* (1998), 315–351.
17. Wiedermann, J. Fuzzy Turing Machines Revised. Tech. Rep. 838, Institute of Computer Science, Academy of Sciences of the Czech Republic, July 2001.
18. Wolkenhauer, O., and Babuška, R. Fuzzy Relational Biology: A Factor-Space Approach to Genome Analysis. In *Computation in Cells: Proceedings of an EPSRC Emerging Computing Paradigms Workshop* (April 2002), Dept. of Computer Science, University of Hertfordshire, UK, pp. 53–58. Technical Report No. 345.
19. Yager, R. R. On the theory of bags. *Int. J. General Systems 13* (1986), 23–37.