Advance Access published on July 20, 2006
doi:10.1093/comjnl/bxl029

Fuzzifying P Systems

APOSTOLOS SYROPOULOS

Greek Molecular Computing Group, 366, 28th October Street, GR-671 00 Xanthi, Greece
Email: asyropoulos@yahoo.com

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. an orthogonal approach to the fuzzification of multisets), 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 promising 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.

Keywords: P systems, fuzzy set theory, computability

Received 8 May 2004; revised 11 May 2006

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 John L. Casti (2003), *The One True Platonic Heaven*: *A Scientific Fiction on the Limits of Knowledge*, Joseph Henry Press, Washington, DC, 2003.

1. INTRODUCTION

P systems [1], 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 bubbles-inside-bubbles structure, where we have a bubble that contains bubbles, which, in turn, contain other bubbles etc., or like the fractal that is shown in Figure 1. Initially, each compartment contains a number of possible repeated objects (i.e. a multiset 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 [2], the standard reference on P systems, for more information.

Fuzzy set theory is a theory that generalizes the concept of the set (for an overview, for example, see [3]). 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 models and methodologies that are based on fuzzy set theory (for example, see [4]). Consequently, the fuzzification of P systems is a quite reasonable development. Indeed, Păun discusses in [2, p. 365] the idea of *approximate* computing in the framework of P systems theory (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.

FIGURE 1. A fractal that resembles the membrane structure of a P system.

In his seminal paper from 1986 [5], Yager introduced *fuzzy multisets*, i.e. structures that can be characterized by higher-order functions $X \rightarrow I \rightarrow \mathbb{N}_0$, where $I = [0, 1]$ and $\mathbb{N}_0$ is the set of natural numbers including zero. (It is not difficult to see that a fuzzy multiset A can also be characterized by a function $X \times I \rightarrow \mathbb{N}_0$.) More specifically, a fuzzy multiset is a structure that consists of multiple copies of objects that belong to the fuzzy multiset to different degrees. However, it is more natural to start with an ordinary multiset and then to define its *fuzzy sub-multisets*, just like we define fuzzy subsets starting from a (crisp) set.

Assume that X is a universe (i.e. a fixed set), then a function $A : X \rightarrow \mathrm{I}$ characterizes any fuzzy subset A of X. In particular, $A(x) = i$ means that x belongs to A to degree i. Suppose that M is a multiset whose support is X. M is characterized by a function $X \rightarrow \mathbb{N}_0$, such that $M(x) = m$ means that M contains m copies of x. Now, it is quite natural to demand that a fuzzy sub-multiset of M is a structure that is characterized by a function $X \rightarrow \mathbb{N}_0 \rightarrow \mathrm{I}$. However, it turns out that this is a really general structure that we will briefly discuss later on. On the other hand, a more natural choice is to have functions $X \rightarrow \mathbb{N}_0 \times \mathrm{I}$ to characterize our fuzzy sub-multisets, which we have dubbed *multi-fuzzy set* to distinguish them from Yager's fuzzy multisets. Thus, given a multiset $M : X \rightarrow \mathbb{N}_0$, such that $M(x) = n$, and a multi-fuzzy set $\mathcal{M} : X \rightarrow \mathbb{N}_0 \times I$, the expression $\mathcal{M}(x) = (n, i)$ will denote that these n copies of x belong to $\mathcal{M}$ to a degree i.

I think this is the right place to briefly discuss a few philosophical issues regarding fuzzy sets, in general. First of all, one should not forget that sets are used to model almost everything. In addition, fuzzy subsets are an extension of the classical notion of a set, which was introduced out of the necessity to model *vagueness*, something so common in our everyday life. For instance, consider a group of people and suppose we have to form a set that will model the subgroup of people that are tall. Clearly, there are no universally accepted criteria that can be used to say that some ordinary Joe is either tall or not tall. For example, if Joe is 1.80 m tall can we say for sure whether he is tall or not? Obviously, if the group of people consists of basketball players, then it is quite possible that Joe is rather short; otherwise, one may say that Joe is rather tall. Certainly, it is far better to say that Joe is tall to degree $i_1 \in \mathrm{I}$ or tall to degree $i_2 \in \mathrm{I}$ instead of saying that Joe is 'rather short' or 'rather tall'. What if our original set was not really a set but rather a multiset. Quite naturally, there are a number of people that have exactly the same height (e.g. Joe and his friend Al are exactly 1.80 m tall), and these people are again 'rather short' or 'rather tall', if we adopt our verbal classification instead of the numerical one. However, one can say that these two men are tall to degree i_1 or i_2 respectively. So, fuzzy sets and mutli-fuzzy sets provide a systematic way to deal with vagueness. But one should be careful and not confuse vagueness with lack of information. When we say that something is fuzzy, we mean that there is no sharp way to distinguish between properties, attributes etc. In addition, probability and fuzziness are two entirely different things. One may say, that a probability is a mathematical characterization of the lack of information regarding the *plausibility* of some phenomenon, event etc. Thus, when we say that Joe is tall with probability i, this clearly indicates lack of information. But we will have the chance to touch this issue again in Section 3.

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 *et al.* [6] have developed the so-called BSS-machine (a sort of Turing machine) that is capable of handling real numbers and real number functions. In addition, Ziegler has examined in [7] the prospects of real number (hyper-)computation in the framework of Type 2 computability theory (see [8] for an overview of Type 2 computability theory). A common characteristic of these computational models is that they treat real numbers as real entities and they transcend the capabilities of the Turing machine. In addition, we should note that Wegner discusses in [6] *super-Turing* computation in the framework of interactive computation, while Kieu advertizes in [10] 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 have led a number of researchers and thinkers to manifest that *hypercomputation* [11] 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 [12] believes that real-number computation does not really go beyond the Church–Turing barrier. On the other hand Ord and Kieu show in [13] that the arguments employed to refute hypercomputation are flawed. And this, of course, is an evidence that the polemics between proponents and opponents of hypercomputation is quite active.

Structure of the paper. We start by formally defining multi-fuzzy sets, possible extensions of the concept and their properties. In addition, we present the theory of fuzzy grammars (FGs) 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 conclude 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 mathematical structure that can be characterized by a function $X \to \mathrm{I} \to \mathbb{N}_0$, where X is some fixed set. Clearly, a fuzzy multiset can be modelled by a function $X \times \mathrm{I} \to \mathbb{N}_0$, where the mapping $(x, i) \mapsto n$ denotes that these n copies of x belong to the fuzzy multiset to 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 [14]) 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 a number of otherwise indistinguishable objects possess a particular property to a certain degree. (Note that the number of indistinguishable objects is called its *multiplicity*.) However, it is quite surprising that *another* number of these objects may belong to a different degree to the same fuzzy multiset. Thus, in order to get out of this awkward situation, we start from a multiset and define its multi-fuzzy (sub-)sets. Let us now proceed with the formal definition of these structures.

DEFINITION 1. *Assume that* $M : X \to \mathbb{N}_0$ *characterizes a multiset* M, *then a multi-fuzzy subset of* M *is a structure* $\mathcal{A}$ *that is characterized by a function* $\mathcal{M} : X \to \mathbb{N}_0 \times \mathrm{I}$, *such that if* $M(x) = n$, *then* $\mathcal{A}(x) = (n, i)$. *In addition, the expression* $\mathcal{A}(x) = (n, i)$ *denotes that the degree to which these n copies of x belong to* $\mathcal{A}$ *is i.*

Obviously, one can go further and extend the definition above. For example, L-multi-fuzzy sets have been introduced in [15].

DEFINITION 2. *Suppose that the set W is a (fixed) universe, L a frame, and* $M : W \to \mathbb{N}_0$ *is a multiset, then an L-multi-fuzzy set is characterized by a function* $X : W \to \mathbb{N}_0 \times L$. The expression $\mathcal{X}(x) = (n, \ell)$ *denotes that* $M(x) = n$ *and that the degree to which these n copies of x belong to X is equal to* ℓ.

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* $A \subseteq X$ *is identical to the multi-fuzzy set* $\mathcal{A}$ *defined as follows*:

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

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

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

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

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

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

DEFINITION 3. *Suppose that* $\mathcal{A}$ *is a multi-fuzzy set having the set X as its universe, then its cardinality, denoted card* $\mathcal{A}$, *is defined as*

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

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 the universe set Z, then their union, intersection, sum and their difference are defined as follows:

DEFINITION 4. (*Union of multi-fuzzy sets*).

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

Notice that in the case of multisets the union is defined in terms of the max operator. Also, the typical definition of fuzzy subset intersection is given in terms of max. Thus, the definition above is fully justified. Similarly, since the operations of set intersection for both multisets and fuzzy subsets are defined in terms of the min operator, the following definition is completely reasonable:

DEFINITION 5. (*Intersection of multi-fuzzy sets*).

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

The sum is an operation that is defined only for multisets and it is actually a generalization of the union, it really makes sense to use the max operator to define the membership degrees, while the multiplicity is defined as usual.

DEFINITION 6. (*Sum of multi-fuzzy sets*).

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

For reasons similar to the previous case, the difference of two multi-fuzzy sets is defined as follows.

DEFINITION 7. (*Difference of multi-fuzzy sets*).

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

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.

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

Clearly, the equality $(\mathcal{A}_\mu + \mathcal{B}_\mu)(z) = (\mathcal{B}_\mu + \mathcal{A}_\mu)(z)$ 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 belong a number of different times to some degree. Here is a formal definition.

DEFINITION 8. *A general multi-fuzzy set is a structure that is characterized by a higher-order function* $X \to \mathbb{N}_0 \to I$. *In other words*, *this is a structure whose elements are multisets that belong to the general multi-fuzzy set to a degree.*

Of course, one can extend the previous definition even further 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 [16] for an overview and [17] 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 extent 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 [18] for details.

In order to understand fuzzy rewriting rules, we need to grasp the notion of a FG. The following definition is a more formal version of the corresponding definition found in [3]:

DEFINITION 9. *A fuzzy grammar, FG, is defined by the quintuple*

$$FG = (V_N, V_T, S, P, A),$$

where

- *V_N is the set of non-terminal 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*

$$A : P \to I.$$

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 non-terminal symbol S.

A string $\alpha \in V_T^*$ is said to belong to the fuzzy language L(FG) if and only if α is a sentential form. In addition, the degree to which α belongs to L(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 k-th derivative and p_i^k denotes the i-th direct derivative in the k-th 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 10. *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 $\rho \in I$ indicate the plausibility that α is reduced to β in a derivation step.

FGs are not just the product of the fuzzification 'storm' mentioned in the introduction. On the contrary, they are quite useful structures as they have found various uses especially in artificial intelligence. Let us give a simple example of their use borrowed from [19]. Suppose we want to construct a system capable of recognizing simple (geometric) shapes in an image. In addition, assume that we have a FG that can be used to draw very simple houses. If a given string s belongs to this grammar with a degree d, then this practically means that we are sure that the drawing generated by s depicts a house with degree equal to d. Thus, an object that appears in an image and identical (?) to the drawing generated by s is recognized as a house with degree equal to d.

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 system 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 belong to a compartment to some degree. Certainly, one may argue that using fuzzy multisets is a more natural choice. However, as we have already explained multi-fuzzy subsets are formed from some multiset and since multisets are the data manipulated by P systems, it is absolutely reasonable to use these structures instead of fuzzy multisets.

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 the n copies of α belong to a designated compartment A is i; also the degree to which the m copies of α belong to a designated compartment B is j. If there is a rule that moves αs 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 αs will be equal to $\max\{i, j\}$ (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:

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

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-mentioned 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, a good reference is [2]).

DEFINITION 11. *A P system with fuzzy data is a construct*

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

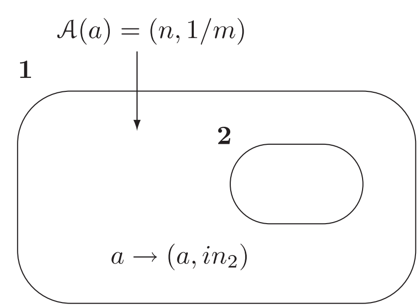
where

(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 injectively labeled with succeeding natural numbers starting with one*;
(iii) *$w^{(i)} : O \to \mathbb{N}_0 \times I$, $1 \leq i \leq m$ 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 $u \to v$, $u \in O^*$ and $v \in O^*_{\mathrm{TAR}}$, where $O_{\mathrm{TAR}} = O \times \mathrm{TAR}$, $\mathrm{TAR} = \{\mathrm{here}, \mathrm{out}\} \cup \{\mathrm{in}_j \mid 1 \leq j \leq m\}$. 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 $i \in \mathbb{N}$.*

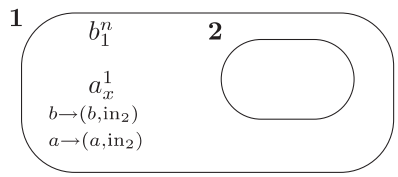
Although Definition 11 is reasonable enough, it is not really clear whether it is necessary to defuzzify the result of the computation. Naturally, it makes sense to go on with the defuzzification process, once we have data that are the result of a fuzzification process. However, it is quite possible 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 their computational power. Naturally, one may view such systems as a form of hypercomputational machines. However, it is too soon to jump into any definitive conclusions. 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. However, there is nothing that prevents one from computing any real number, if we assume that objects may have real numbers as membership degrees. Indeed, the following theorem makes this explicit.

THEOREM 2. *P systems with fuzzy data can compute any positive real number.*

Proof. Assume that $y \in \mathbb{R}_0^+$ (i.e. the set of all positive reals including zero), then $y = x + n$, where $x \in [0, 1]$ and $n \in \mathbb{N}$. Let us now construct the following P system.



Notice that α_x^n means that there are n copies of α that belong to the compartment to degree equal to x. When this P system will halt, the cardinality of the multi-fuzzy set that is contained in compartment 2 will be equal to y. Thus, trivially, P systems such as this can compute any real number.

□

In addition, it is quite possible to compute even negative real numbers, if we opt to have a P system with a fuzzy version of the so-called *hybrid* sets [20]. However, one has to justify the use of these structures and also to carefully investigate the properties of the resulting systems.

Skeptic readers may doubt whether P systems with fuzzy data (i.e. systems whose compartments are populated with multi-fuzzy sets) can actually compute any real number at all. In particular, these skeptic readers may argue that by associating a number (i.e. a membership degree) to a group of identical objects, one does not get a concrete way to represent a number. Before going on, I believe it is rather important to note that a Turing machine or a P system performs a computation just because we, as external observers, are inclined to perceive their operation as computation. In addition, it is up to us to associate the final state of a 'computing' device to a number, which is the outcome of the computation. In the case of P systems with fuzzy data, the input data are represented by a multi-fuzzy set and the output is the cardinality of a multi-fuzzy set. In other words, the multi-fuzzy set contained in the output compartment represents a number. Whether we can 'translate' this number into a familiar notation is an entirely different issue, which will not concern us here.

Another objection that may be posed is that one cannot have as input and, consequently, as output irrational numbers. First of all, one should notice that there are at least three irrational numbers that affect our lives in many different, and sometimes profound, ways. These numbers are the number π, the number e and the number ϕ (the golden ratio). In particular, π appears in many equations describing fundamental principles of the universe. For instance, Coulomb's law for the electric charge

$$F = \frac{|q_1 q_2|}{4\pi\epsilon_0 r^2}$$

and Heisenberg's uncertainty principle

$$\Delta x \Delta p \geq \frac{h}{4\pi}$$

are such equations. Clearly, in physics we work with approximations, but, on the other hand, these equations describe real phenomena and, of course, π is an important part of the description of these phenomena in spite of the fact that we *cannot* represent it by conventional means. On the other hand a circle with radius 1 unit is a faithful representation of 2π. Similarly, one can say that the multi-fuzzy set contained in the output compartment after the operation of the system has ceased is a representation of π or, for that matter, of any irrational 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. But we repeat and emphasize that it is too early for such a definitive conclusion. On the other hand, it is an open problem whether P systems with fuzzy data, which fully interact with their environment, are really capable of hypercomputation.

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 multiset rewriting rules differ fundamentally from P systems with probabilist rewriting rules in that there is no bias in the selection of the rules.

DEFINITION 12. *A P system with fuzzy multiset rewriting rules is a construct*

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

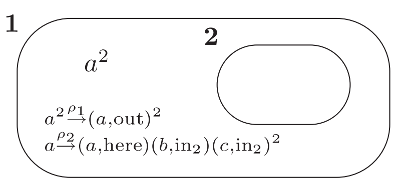
where each $\sigma_i : R_i \to I$ is a fuzzy set defined over R_i, $\sqcap$ is a computable t-norm and $\sqcup$ a computable t-conorm. All other components are identical to those of an ordinary P system.

In the previous definition, we demanded 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 that there is an *effective* procedure[1] by means of it we can compute their values for all possible arguments.

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 13. *Assume that in the end of the computation the output compartment* i_0 *contains copies of the objects* $\beta_1, \beta_2, \ldots, \beta_n$. *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 rewriting rules:*



Suppose that this system halts after n steps, then the crisp result of the computation will be equal to 6*n. Now, the degree to which* 6*n 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* [2. *p*. 56].

Fuzzy Turing machines (for example, see [21, 22]) 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 [23]:[3]

DEFINITION 14. *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 states 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 $\bigvee$ (i.e. the smallest fuzzy relation that contains $\bigvee$ and is transitive).

THEOREM 3. *For every P system with fuzzy multiset rewriting rules there is a fuzzy Turing machine that computes exactly the same set of numbers.*

Sketch 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 Turing machines. In particular, this class of P systems includes system transition P systems and P systems with cooperating rules, systems with bi-stable catalysts, systems with plain or bi-stable catalysts and

[1] I will not make any attempt to provide a precise definition of the term 'effective'. I will assume the 'usual' meaning of the word.

[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 $(b \oplus (c) \otimes a = (b \otimes a) \oplus (c \otimes a)$.

[3] A referee has pointed out that Santos wrote his important paper in 1970, while t-norms and t-conorms gained widespread acceptance in the 80s. Thus, if Santos would have known t-norms and t-conorms he would not had used semirings.

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 latter 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.

In general, it has been proved that P systems have the computational power of Turing machines [2]. From this, it is not difficult to see that for every fuzzy Turing machine there is a P system that computes exactly the same numbers. However, one should notice that cells respond to external stimuli in a number of, sometimes unexpected, ways. In addition, it has been shown that interactive computing systems are more powerful than Turing machines [9]. Thus, if we allow P systems to freely interact with their environment (e.g. by replacing multiset rewriting rules with ‘stream rewriting’ rules that affect the number of elements in any given compartment), then one might expect to get conceptual computing devices that are more powerful than Turing machines. Needless to say is that if the previous expectation turns out to be true, the fuzzy version of such P systems will be more powerful than fuzzy Turing machines. Clearly, these are just speculation, but we hope to have concrete results in the near future (see [24] for an overview of the work that has been done so far).

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 observed that 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.

ACKNOWLEDGEMENTS

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

REFERENCES

[1] Păun, G. (2000) Computing with Membranes. *J. Comput. Syst. Sci.*, **61**, 108–143.

[2] Păun, G. (2002) *Membrane Computing: An Introduction*, Springer-Verlag, Berlin.

[3] Klir, G. J. and Yuan, B. (1995) *Fuzzy Sets and Fuzzy Logic: Theory and Applications*, Prentice Hall, Englewood Cliffs, NJ.

[4] Wolkenhauer, O. and Babuška, R. (2002) Fuzzy relational biology: a factor-space approach to genome analysis. In *Computation in Cells: Proc. EPSRC Emerging Computing Paradigms Workshop*, Department of Computer Science, University of Hertfordshire, UK. pp. 53–58, Technical Report No. 345.

[5] Yager, R. R. (1986) On the theory of bags. *Int. J. Gen. Sys.*, **13**, 23–37.

[6] Blum, L., Cucker, F., Shub, M. and Smale, S. (1997) *Complexity and Real Computation*. Springer-Verlag, New York.

[7] Ziegler, M. (2005) Computability and continuity on the real arithmetic hierarchy and the power of Type-2 nondeterminism. In Cooper, S. B., Löwe, B. and Torenvliet, L. (eds), *Proc. New Computational Paradigms: First Conf. on Computability in*

Europe, CiE 2005, Amsterdam, The Netherlands, June 8–12, *LNCS* **3526**, pp. 562–571. Springer-Verlag, Berlin.

[8] Weihrauch, K. (2000) *Computable Analysis: An Introduction*, Springer-Verlag, Berlin.

[9] Wegner, P. (1998) Interactive foundations of computation. *Theor. Comput. Sci.*, **192**, 315–351.

[10] Kieu, T. D. (2002) Quantum hypercomputation. *Minds Mach.*, **12**, 541–561.

[11] Syropoulos, A. (2006) *Hypercomputation: Computing Beyond the Church-Turing Barrier*. Springer, New York.

[12] Cotogno, P. (2003) Hypercomputation and the physical Church–Turing thesis. *B. J. Philos. Sci.*, **54**, 181–223.

[13] Ord, T. and Kieu, T. D. (2005) The diagonal method and hypercomputation. *Br. J. Philos. Sci.*, **56**, 147–156.

[14] Miyamoto, S. (1996) Fuzzy multisets and application to rough approximation of fuzzy sets. *Proc. Fourth Inter. Workshop on Rough Sets, Fuzzy Sets, and Machine Discovery (RSFD'96)*, University of Tokyo, Tokyo, Japan. pp. 255–260.

[15] Syropoulos, A. (2006) On generalized fuzzy multisets and their mechanical processing. Submitted for publication.

[16] Obtułowicz, A. and Păun, G. (2003) (In search of) Probabilistic P systems. *BioSystems*, **70**, 107–121.

[17] Madhu, M. (2003) Probabilistic rewriting P systems. *Int. J. Found. Comput. Sci.*, **14**, 157–166.

[18] Klir, G. and Folger, T. (1987) *Fuzzy Sets, Uncertainty, and Information.*, Prentice Hall, Englewood Cliffs, NJ.

[19] Stanchev, P. L. and David, G.Jr. (2002) Formal languages for image primitive description. *Infor. Theor. Appl.*, **9**, 51–60.

[20] Loeb, D. (1992) Sets with a negative number of elements. *Adv. Math.*, **91**, 64–74.

[21] Moraga, C. (1999) Towards a fuzzy computability? *Mathw. Soft Comput.*, **6**, 163–172.

[22] Wiedermann, J. (2001) *Fuzzy Turing Machines Revised*, Technical Report 838. Institute of Computer Science, Academy of Sciences of the Czech Republic.

[23] Santos, E. S. (1970) Fuzzy Algorithms. *Inform. Control*, **17**, 326–339.

[24] Syropoulos, A. and Doumanis, S. (2006) On C systems. Unpublished manuscript.