Computing in a Fuzzy Environment

Apostolos Syropoulos

366, 28th October Str., GR-671 00 Xanthi, GREECE. Email: asyropoulos@aol.com
Web-page at http://obelix.ee.duth.gr/~apostolo

Fuzzy set theory is a mathematical theory about vagueness, which is a fundamental propery of this world. Some believe that fuziness is a just one facet of vagueness, but some other believe that it is the only expression of vagueness. Despite of this, most models of computation ignore vagueness and "exist" in an exact and defectless world. And this assumption has affected the way real computers are being built. By introducing fuzziness to models of computation, one aims to develop models of computation that are closer to *reality*. The text that follows is a brief survey of some fuzzy models of computation.

Originally, the word computing was synonymous with counting and reckoning, and a computer was an expert at calculation. In the 1950s with the advent of the (electronic) computer, the meaning of the word computing was broadened to include the operation and use of these machines, the processes carried out within the computer hardware itself, and the theoretical concepts governing them. These theoretical concepts find their roots in the Turing machine [1], that is, a conceptual computing device that was devised by Alan Turing, the great British logician, in the mid 1930s. In essence, Turing showed that computation is a process of symbol manipulation, since his machine is dully processing symbols that are printed on a tape. More specifically, a Turing machine consists of paper tape divided into cells and a scanning head that moves across the tape and can read and write symbols on each cell. At any given moment, a machine is in a state. Depending on the state the machine is and the contents of the cell that is currently being scanned by the scanning head, the machine enters a new state and either replaces the contents of the cell or moves to the cell that is either to the left or to the right of the cell being scanned. At each moment, the machine consults the controlling device to determine what to do next. By using a technique, which was developed by Kurt Gödel, the famous Austrian mathematician, Turing had shown that it is possible to "construct" a universal machine, which would take as input the specification of a machine together with its input, that is capable of solving a great number of problems. The importance of this universal machine is so great that some thinkers claim that modern computers are in fact realizations of it. However, this claim is clearly an exaggeration as modern computers are able to interact whereas Turing machines do not interact with their environment.

Despite the fact that the Turing machine is not the only model of computation, still it is the most widely known model. There are other models that equally interesting. For example, P systems [2], which have been proposed by Gheorghe P un, is a model of computation inspired by the way living cells function. Basically, a P system is structure that consists of nested, porous membranes that contain indistinguishable copies of objects. Attached to each compartment is a set of rewrite rules, that is, equations that roughly specify how the contents of a compartment should be modified (strictly speaking, a rewrite rule is a method to transform a character string into a new one; for example, the Unix sed utility is a program that allows users to implement simple rewrite rules). In particular, such rules may specify that copies of certain objects should be deleted or moved to another compartment or that copies of objects should be introduced from outside or be created out of thin air. Rules are applied in parallel in such a way that only optimal output is generated. When there is no more activity, the result of the computation is equal to the number of (copies of the) objects found in a designated compartment—the output compartment. P systems operate in a massively parallel way while they can interact with their environment.

A basic assumption of both Turing machines and P systems is that all operations are exact. In different words, vagueness is taken seriously into account. It is true that there are probabilistic versions of both Turing machines and P systems, but this is not the kind of vagueness that offers a new insight. On the other had, although fuzziness is basic expression of vagueness, still its use in models of computation is largely ignored by most computers scientists. But what exactly is fuzziness?

Fuzzy set theory was developed by Lotfi A. Zadeh [3] who had the ingenious idea to define an extension of the notion of a set where elements belong to a degree. In particular, given a collection of objects, one can assign a membership degree, which is usually a number greater or equal to zero or less than or equal to one, to each object to form a fuzzy subset of this collection. As a concrete example, consider a group of people. Then we can form the fuzzy subset of tall people of this group. Depending on the height of the people and our knowledge of the height of people, in general, one can form this fuzzy subset. In the most general case, one can argue that the membership degrees are not algorithmic in nature. On the contrary, one could say that they are based on the subjective judgement of some expert.

Fuzzy sets have been successfully used in various applications and that is why they are particularly "popular" among engineers. However, not everyone shares this enthusiasm. Indeed, there are many thinkers who believe that statements like *Serena is tall to a degree of 0.70* are basically elementary mistakes of logic. These thinkers propose that statements like this can be rephrased as *Serena is 70% tall* and in this case the statement is either true or false In the first statement we actually say that *Serena is tall* and the truth degree of this statement is 0.70, where 1 denotes absolute truth. The interesting thing is that one can use statements like this in (fuzzy) inference rules to make deductions, something not possible with the exact statement above. In other words, the real difference between the two approaches is in the denotation of the two statements.

Zadeh believes that probability theory and fuzzy set theory are rather complementary in that one can use fuzzy sets in cases where probability theory is not useful and vice versa. Others believe that fuzzy set theory is more fundamental than probability theory and randomness. Interestingly, the recent proof of the "Free Will Theorem" by John Horton Conway and Simon Kochen has revealed that *[I] f experimenters have a certain property, then spin 1 particles have exactly the same property. Since this property for experimenters is an instance of what is usually called "free will" we find it appropriate to use the same term also for particles* [4, p. 1444]. Interestingly, they derived their result without using probability theory or randomness, while they conclude that randomness is not needed. But this does not exclude vagueness as a property of this world, thus, fuzziness can be used to reason about vague things. These remarks make it clear that models of computation based on fuzzy set theory are closer to *reality* than anything else. In the rest of this article, I will briefly present fuzzy Turing machines and fuzzy P systems, which are the most promising models of fuzzy computation.

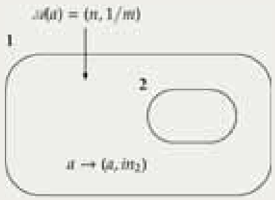
Fuzzy Turing Machines Eugene S. Santos [5] is the first researcher who formalized fuzzy Turing machines. However, it was Zadeh that spoke first about fuzzy algorithms [6]. In the most general case, a fuzzy Turing machine is one where the transition from one configuration (i.e., the state the machine is together with the information regarding the cell that is being scanned and the information contained in the cell) to another configuration (i.e., the new state that the machine enters, what has been written in the cell and which direction the scanning head has moved) is associated with truth degree. In the end, a fuzzy Turing machine computes a result with some plausibility degree. The important question concerning fuzzy Turing machines is whether they can solve problems that their non-fuzzy counterparts cannot. Indeed, there are problems that cannot be solved using Turing machines. For example, due to Turing's work on the halting problem, we know that when a program gets stalled, a computer program cannot say whether it has entered an infinite loop or not. Now, Jiří Wiedermann [7], has shown that fuzzy Turing machines can solve problems that cannot be solved by an ordinary Turing machine. In particular, it seems that fuzzy Turing machines can decide whether problems like Goldbach's conjecture is true or not. This conjecture asks whether every even integer greater than 2 is a Goldbach number, that is, a number that can be expressed as the sum of two primes. Naturally, one may ask why we do not use them to solve such problems. The answer is very simple: we need to program the machine to solve the problem! Thus, what Wiedermann claims is that we can program the machine to solve the problem but he does not offer a solution to any particular problem. Stricktly speaking, Wiedermann claims that fuzzy Turing machines are actually hypermachines [8], that is, machines that can solve problems no Turing machine can solve.

Fuzzy P Systems Since vagueness is basic property of our world, it makes sense to expect biological systems to be vague. Thus, a fuzzy version of P systems is model of computation that is closer to reality. Fuzzy P systems have been introduced by this author [9]. In order to explain how these systems operate, it is necessary to say a few things about *fuzzy multisets* and fuzzy rewrite rules.

A *multiset* [10] is as generalization of the concept of a set. While ordinary sets contain different elements, a multiset contains copies of different elements. For example, {1,1,2,2,2,3,4,4,,4} is a multiset that contains two copies of "1," three copies of "2," one copy of "3," and three copies of "4." Typically, it is assumed that the number of copies is finite. Fuzzy multisets have been introduced by Ronald R. Yager [12]. In fact they are generalization of both ordinary sets and multisets—different number of copies of elements may belong to different degrees. This means, that it is possible to have three copies of element "x" where each of them belongs to the set with degree 0.62 and seven copies of "x" that belong to the set with degree 0.81! Although this may seem weird, still it is quite possible to find real world examples where this makes sense. Nevertheless, for our purpose we need a restricted form of fuzzy multisets, that I have coined *mutli-fuzzy sets,* where it is possible to have only one set of copies for each element that, naturally, belong to the set to a degree. Usually, the term *cardinality* refers to the number of elements of some set. In the case of a multi-fuzzy set, the cardinality is the sum of all element *occurrences*, where for each element the occurrence is the product of the number of copies of this element times its membership degree. Now, a fuzzy rewrite rule is one that it is possible (not probable!) to transform a character string into another one with a specific plausibility degree.

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 a belong to a designated compartment A is i; also the degree to which the m copies of a belong to a designated compartment B is j. If there is a rule that moves as 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 as 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. P systems with fuzzy data produce, in general, real positive numbers and so, unexpectedly, extend their computational power.

The following picture depicts a simple example of P system with fuzzy data:



This P system contains n objects in compartment 1, which will be transferred into compartment 2. 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. *P systems with fuzzy data can compute any positive real number.*

By replacing the ordinary rewrite rules with fuzzy rewrite rules, a result is computed with some plausibility degree and this makes things even more interesting. The computational power of these machines is an open problem.

Given a fuzzy set and element may belong to it with degree μ while it does not belong to the set with degree that is equal to $1-\mu$. Krassimir T. Atanassov [13] has proposed an interesting extension to fuzzy sets: sets where the non-membership degree is an "arbitrary" number ν such that $0 \leq \mu + \nu \leq 1$. These structures are known as *intuitionistic* fuzzy sets, but the term intuitionistic is a misnomer since these structures have nothing to do with intuitionistic mathematics. Clearly, one can define the corresponding extension of multi-fuzzy sets and show their properties (see [14] for details). This author has defined P systems that are populated with "intuitionistic" multi-fuzzy sets [15]. It is possible to go further in the generalization ladder to define really general multi-fuzzy sets and fuzzy P systems (see [17] for more details).

Apart from fuzzy models of computation, one can define fuzzy models of concurrency that better explain what happens in reality. For example, processes in a system can be modeled by fuzzy multisets, where the membership degree denotes the degree to which a process is similar to a prototype process that consumes minimal resources. Next, one can define fuzzy *evolution* rules that "specify" how a system evolves (see [18] a presentation of this idea).

Epilogue Despite its age, the field of fuzzy computing is relatively immature since many ideas have been developed the last few years. This means that students and researchers who are looking for a new exciting research should consider doing a research project in fuzzy computing!

References

[1] Turing, A. M. On Computable Numbers, with an application to the Entscheidungsproblem. Proceedings of the London Mathematical Society 42 (1936), 230–265.

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

[3] Zadeh, L. A. Fuzzy Sets. Information and Control 8 (1965), 338–353.

[4] Connway, J. H., and Kochen, S. The Free Will Theorem. Foundations of Physics 36, 10 (2006), 1441–1473.

[5] Santos, E. S. Fuzzy Algorithms. Information and Control 17 (1970), 326–339.

[6] Zadeh, L. A. Fuzzy Algorithms. Information and Control 12 (1968), 94–102.

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

[8] Syropoulos, A. Hypercomputation: Computing Beyond the Church-Turing Barrier. Springer New York, Inc., Secaucus, NJ, USA, 2008.

[9] Syropoulos, A. Fuzzifying P Systems. The Computer Journal 49, 5 (2006), 619–628.

[10] Syropoulos, A. Mathematics of Multisets. In Calude et al. [10], pages 347–358.

[11] Cristian S. Calude, Gheorghe P un, Grzegorz Rozenberg, and Arto Salomaa, editors. Multiset Processing, number 2235 in Lecture Notes in Computer Science, Berlin, 2001. Springer-Verlag.

[12] Yager, R. R. On the theory of bags. Int. J. General Systems, 13 (1986), 23–37.

[13] Atanassov, K. T. Intuitionistic fuzzy sets. Fuzzy sets and Systems 20, 1 (1986), 87–96.

[14] Syropoulos, A. On Nonsymmetric Multi-Fuzzy Sets. Critical Review IV(2010), 35–41.

[15] Syropoulos, A. Intuitionistic Fuzzy P Systems, submitted for publication 2010.

[16] Syropoulos, A. On Generalized Fuzzy Multisets and their Use in Computation, submitted for publication 2010.

[17] Syropoulos, A. On Generalized Fuzzy Multisets and their Use in Computation. A preliminary version is available form http://ppage.psystems.eu/index.php/Papers

[18] Syropoulos, A. Fuzzy Chemical Abstract Machines, arXiv:0903.3513v1 [cs.FL], 2009.