94

Vague Computing Is the Natural Way to Compute!

Apostolos Syropoulos

94.1 Introduction

Typically, when a computer performs a task, it can be seen as a calculation or a reckoning. For example, consider a simple arcade video game where the machine continuously gets input from the user and computes the new position of some "characters" that move on a board, etc. A particularly interesting aspect of computation is that the majority of people understand it as an exact function. Nevertheless, this is an excessive expectation or requirement, depending on how one perceives computation. In particular, users expect computer programs to deliver exact results while computer programmers work under the assumption that everything is precisely defined and no vagueness arises anywhere. But is this a reasonable assumption?

The answer to this question is not a simple one. For instance, we are successfully using computers that operate in a precise manner for many years and we have achieved much with these devices. Or is this an oversimplification of what actually happens in reality, whatever this may mean? Obviously, digital computers execute software in the expected way as long as hardware operates within some tolerance range. So one may be tempted to say that everything related to computers is based on an illusion or a rough assumption. One may argue that this is an exaggeration, nevertheless, it is a view that may help us understand things differently.

Vagueness is widely accepted to characterize terms that, to some extent, have borderline cases, that is, a case in which it seems impossible either to apply or not to apply a vague term. The Sorites Paradox, which was introduced by Eubulides of Miletus, is a typical example of an argument that shows what it is meant by borderline cases. The term sorites derives from the Greek word soros, which means "heap." The paradox is about the number of grains of wheat that make a heap. All agree that a single grain of wheat does not comprise a heap. The same applies for two grains of wheat, three grains of wheat, etc. However, there is a point from which the number of grains becomes large enough to be called a heap, but there is no general agreement as to where this occurs, hence the paradox.

In general, there are everyday objects and activities that seem to be exact, yet they are vague! For example, "[e]xperience has shown that no measurement, however carefully made, can be completely free of uncertainties" [15, p. 3]. Remarks like this one may have some "unexpected" consequences. For instance, one might go as far as to argue that vagueness is the norm and exactness the exception! If this is not an exaggeration, which is not as I will show later on, then one could reasonably argue

R. Seising et al. (Eds.): *On Fuzziness: Volume 2*, STUDFUZZ 299, pp. 653–658.
DOI: 10.1007/978-3-642-35644-5_94

that many, if not most, things are vague by definition. Thus, one should be able to employ vagueness in computation or, even, she should to be able to perform truly *vague* computations, whatever that may mean.

94.2 What Is Vagueness?

Bertrand Russell [9] was perhaps the first thinker who had given a definition of vagueness: "*Per contra*, a representation is *vague* when the relation of the representing system to the represented system is not one-one, but one-many." According to this view, a photograph that is so smudged that it might equally represent three different persons is vague. Building on Russell's ideas Max Black [2] had argued that most scientific theories, computability theory included, are "ostensibly expressed in terms of objects never encountered in experience." Black [2] proposed as a definition of vagueness the one given by Charles Sanders Peirce: "A proposition is vague when there are possible states of things concerning which it is intrinsically uncertain whether, had they been contemplated by the speaker, he would have regarded them as excluded or allowed by the proposition. By intrinsically uncertain we mean not uncertain in consequence of any ignorance of the interpreter, but because the speaker's habits of language were indeterminate." According to Black, the word *chair* demonstrates the suitability of this definition. But it is the "variety of applications to objects differing in size, shape and material" that "should not be confused with the vagueness of the word." In different words, vagueness should not be confused with *generality*. Russell and Black had argued against this misconception. A term or phrase is *ambiguous* if it has at least two specific meanings that make sense in context. Thus, one should not confuse ambiguity with vagueness.

It is widely accepted that there are three different expressions of vagueness [11]:

Many-Valued Logics and Fuzziness. Borderline statements are assigned truth- values that are between absolute truth and absolute falsehood. In the case of fuzziness, truth-values are usually drawn from the unit interval.

Supervaluationism. The idea that borderline statements lack a truth value.

Contextualism. The truth value of a proposition depends on its context (i.e., a person may be tall relative to American men but short relative to NBA players).

94.3 From Exact Computing to Fuzzy Computing

Conceptual computing devices are idealizations of tools that can perform computations. However, these idealizations tend to overlook details concerning the process of computation. This is exactly where vagueness, in general, and fuzziness, in particular, comes into play. I will try to be more specific by presenting two exact models of computation, namely Turing machines and P systems, and how one can easily fuzzify these models. Let me start with Turing machines, which are considered to be the archetypal model of computation.

Turing machines were introduced by Alan Mathison Turing [16] in order to give a formal definition of the notion of computation. In addition, the machine was used in order to give an answer to the *entscheidungsproblem* posed by David Hilbert (i.e., a problem that can be answer with yes or no, in different words a decision problem). Typically, a Turing machine consists of an infinite tape, a controlling device, and a scanning head. The tape is divided into an infinite number of cells. The scanning head can read and write symbols in each cell. The symbols are elements of some set Σ. At any moment, the machine is in a state q_i, which is a member of a finite set Q. What should happen next depends on the symbol just read and the current state and this is hardwired into the controlling device. If no action has been specified for a particular combination of state and symbol, the machine halts. Tuples that conditionally describe the next action are called configurations.

At this point, it is rather interesting to note that Carole E. Cleland [3] has concluded that "Turing machines may be characterized as providing procedure schemas, i.e., temporally ordered frameworks for procedure." In addition, she has claimed that "Turing machine instructions cannot be said to prescribe actions, let alone *precisely* describe them." Based on these one could argue that Turing machines are not computing devices. Surely, this is an exaggeration, nevertheless, it clearly shows that this model of computation is not as well-thought-of as it was always considered to be. Furthermore, Cleland has argued against the idea that Turing machine "symbols" are genuine symbols.

Such remarks and conclusions clearly show that the Turing machine model of computation is implicitly vague. Thus, it does make sense to explicitly introduce vagueness into this model. Indeed, first Lotfi A. Zadeh [17] *vaguely* described a fuzzy Turing machine where configurations form a fuzzy subset. Based on Zadeh's ideas, Eugene S. Santos [10] had formally defined fuzzy Turing machines. The evolution of fuzzy Turing machine, in particular, and fuzzy computing devices, in general, is described in a forthcoming book by this author [13].

P systems is a model of computation that was introduced and popularized by Gheorghe Păun [8]. P systems are conceptual computing devices made up of nested compartments surrounded by porous membranes that define and confine these compartments. Initially, each compartment contains a number of possibly repeated objects, that is, a multiset of objects. When "computation" commences, compartments exchange objects according to a number of multiset processing rules that are associated with each compartment. The activity stops when no rule can be applied. The result of the computation is equal to the number of objects that reside in a designated compartment called the *output membrane*.

As in the case for Turing machines, one can easily see that vagueness is part of this machinery. First, one can never be sure that membranes contain exact copies of some object—it is more reasonable to expect copies to be similar. Also, one may argue that the rules should not be exact, but should give an "outline" of what may happen. These and other aspects of P systems have been studied by this author [12, 14].

94.4 The Need for Fuzzy Computing

Unfortunately, the notion of *uncertainty* is considered by many to be almost equivalent to vagueness, which, of course, is wrong. This is one reason why there is a debate over the superiority of either fuzzy set theory or probability theory to represent vagueness. Clearly, this debate is far from settled. Basically, there are three views—one that naturally claims that fuzzy set theory has nothing new to offer, one that advocates that fuzzy sets and probabilities are two facets of uncertainty (e.g., see [18]), and one that assumes that fuzziness is a fundamental property of our world. There is no question that the first view is deeply flawed. The second view is also problematic, since it considers vagueness and uncertainty to be the same thing. The third view, is, in my eyes, the most reasonable approach. In particular, when I say more fundamental, I mean that most, if not all, natural processes can be characterized as vague, while probabilities are "theoretical quantities which, once the sets and the measure functional on those sets are chosen ('the model'), are capable of being calculated exactly and are perfectly definite (real) numbers which contain no reference to chance" [4, p. 45]. Last, but certainly not least, Bart Kosko [7] has also convincingly argued that fuzzy set theory is more fundamental than probability theory.

One reasonable question that may pop on one's mind is the following: If vagueness is a fundamental property of our world, how should this affect the way we compute? First, let me stress that until now vagueness was not taken under consideration by any computing device. Engineers have employed various techniques in order to ensure that a "digital logic" is correctly implemented, yet they did so using *vague* constituents! Next, one could argue that just like probabilities are employed in ordinary (aka crisp) computer programs, one could analogously use fuzziness in crisp programs. Indeed, one can implement fuzzy databases, fuzzy programming languages, etc. [6]. Nevertheless, this approach has a serious drawback—it implicitly implies that vague computing can be implemented in machines that operate in an "exact" manner. So, if vagueness is a fundamental property of our world, why should we add an artificial layer to perform vague computational tasks? The answer is not easy, but the reason for this apparent disparity lies in the way we have learned to think. From ancient times, people tried to think in terms of pure and precisely defined objects. In addition, simple things such as reckoning were considered precise operations. For example, two plus two equals four since when one has two sheep and gets two more sheep, she has four sheep in the end. However, what happens when one exchanges two really well-fed animals with two starving animals? In principle, she still has four animals, but they are not the same! Thus, one can argue that even arithmetic is the result of an oversimplification. In different words, exactness should be considered as a limit case and vagueness the norm and not the other way around!

From the discussion so far one may wrongly deduce that there is no fuzzy hardware when, in fact, a good number of researchers is working in the design and construction of real fuzzy hardware.[1] Although there is fuzzy hardware, there is nothing

[1] For example, see [5] for a not so up-to-date account of fuzzy hardware.

that can be classified as a general purpose fuzzy computer. Nevertheless, it is more than necessary to build such a machine in order to be able to fully exploit vagueness in computing. I expect that such machines will be able to solve more easily *everyday* problems that concern *ordinary* people. Since such machines should be equipped with the analog of an operating system and the corresponding tools for programming, editing, etc., more research on fuzzy programming and computing should be carried out. For instance, the work on the definition of a fuzzy version of the λ-calculus by Daniel Sánchez Alvarez and Antonio F. Gómez Skarmeta [1] can be seen as step towards this goal.

Fig. 94.1. Artistic impression of a fuzzy computer. Original drawing by Nikos Amiridis; post-processing with gimp by Apostolos Syropoulos.

References

1. Sánchez Álvarez, D., Gómez Skarmeta, A.F.: A Fuzzy Language. Fuzzy Sets and Systems 141, 335–390 (2002)
2. Black, M.: Vagueness. An Exercise in Logical Analysis. Philosophy of Science 4(4), 427–455 (1937)
3. Cleland, C.E.: On Effective Procedures. Minds and Machines 12, 159–179 (2002)
4. Cook, D.B.: Probability and Schrödinger's Mechanics. World Scientific, Singapore (2002)
5. Kandel, A., Langholz, G. (eds.): Fuzzy Hardware: Architectures and Applications. Kluwer Academic Publishers, Dordrecht (1997)
6. Klir, G.J., Yuan, B.: Fuzzy Sets and Fuzzy Logic: Theory and Applications. Prentice Hall (Sd) (1995)
7. Kosko, B.: Fuzziness vs. Probability. International Journal of General Systems 17(2), 211–240 (1990)
8. Păun, G.: Membrane Computing: An Introduction. Springer, Berlin (2002)
9. Russell, B.: Vagueness. Australasian Journal of Philosophy 1(2), 84–92 (1923)
10. Santos, E.S.: Fuzzy Algorithms. Information and Control 17, 326–339 (1970)

11. Sorensen, R.: Vagueness. In: Zalta, E.N. (ed.) The Stanford Encyclopedia of Philosophy, Fall 2008 edition (2008)
12. Syropoulos, A.: Fuzzifying P Systems. The Computer Journal 49(5), 619–628 (2006)
13. Syropoulos, A.: Fuzzy Computability. Book in preparation (2012)
14. Syropoulos, A.: On Generalized Fuzzy Multisets and their Use in Computation. To appear in the Iranian Journal of Fuzzy Systems (2012)
15. Taylor, J.R.: An Introduction to Error Analysis: The Study of Uncertainties in Physical Measurements, 2nd edn. University Science Books, Sausalito (1997)
16. Turing, A.M.: On Computable Numbers, With an Application to the Entscheidungsproblem. Proceedings of the London Mathematical Society 42, 230–265 (1936)
17. Zadeh, L.A.: Fuzzy Algorithms. Information and Control 12, 94–102 (1968)
18. Zadeh, L.A.: Discussion: Probability Theory and Fuzzy Logic Are Complementary Rather Than Competitive. Technometrics 37(3), 271–276 (1995)