Daoism and Computation

Carl M. Johnson

Contents:

I. Introduction

In order to better understanding the meaning of *daode*[1] implicit in the *Daodejing* 道德經 (also called *Laozi* 老子), it is useful to offer a contrasting understanding of a contemporary paradigm that bears a notable, if superficial, similarity to it, namely that of computation. In recent years, familiarity with the computational paradigm has made it more natural to draw certain analogies between the working out of *daode* and the computing of an algorithmic process. These analogies become even more interesting when one considers the suggestions (as in Stephen Wolfram's *A New Kind of Science* and Gregory Chaitin's *Meta Math!* recently and presaged by John Wheeler's "It from bit" concept and Konrad Zuse's *Calculating Space*)[2] that our world might be fundamentally computational and when one compares these suggestions to the process ontology often associated with contemporary Daoist studies.

---

[1] This essay concerns the *daode* 道德 rather than "the Dao," since in Daoist thought, a field (*dao*) is not experienced outside of its particular focii (*de*). Indeed, Daoism in the first instance was first known as the school of "daode," and only later became "Daoism" without reference to *de*. Sima Tan in the Records of the Grand Historian (later completed by his son Sima Qian) was the first to designate the Six Schools, one of which he called the school of "Daode." See Shiji 世紀 (Shanghai: Zhonghua, 1959), p. 3118.

Computation and the process interpretation of Daoism do share certain similarities, not the least of which is that in both computation and *daode*, one system serves as the microcosmic simulation or representation of another. However, a computational view of *daode*, while helpful as a starting point for understanding *daode*, ultimately overlooks the fundamental differences between the mechanical nature of the extension of computation (the possibility of implicit non-determinisms within conventional computational system not withstanding) and the creative nature of *daode*, which emerges from the encompassing nature of *dao* and the particularizing nature of *de*.

In order to properly make our evaluation, we must first begin with an appropriate definition of computation. With this definition in place, we can then use a case study to point out the many correspondences between the computational modeling and the Daoist modeling, before ultimately refining our understanding of *daode* through contrasting its inclusive, creative totality of insistent particulars with the privileged perspectives implicitly assumed by the computational model.

II. Definition of computation

We begin by noting a shortcoming in much of the existing literature concerning computation. Very often when one examines a seminal work in computational theory—eg. Alan Turing's *On Computable Numbers*—one observes that the author defines computation only in terms of the theoretical capabilities of a particular kind of mechanical procedure and not computation in general. Though mathematicians and computer scientists have shown no reluctance in defining new and different forms of computers and computational procedures in order to explore their limitations, nevertheless, it is

---

[2] Although, *A New Kind of Science* has had relatively little impact outside of the promotion of Wolfram himself and *Meta Math!* is a popularizing turn rather than a scholarly work, I take them as emblematic of a new kind of thinking that is increasingly prevalent. In one sense, the computational model of the world is no different from the once popular Laplacean determinism except in terms of its emphasis on iterative calculation over algebraic continuities. Nevertheless, given the prominence of this kind of thinking, the author considers it to be worth considering the claims of "digital physics" in ontology to be a serious parallel movement to computational functionalism in the theory of mind.

difficult to find a definition of computation in general. In part, this may be because it is widely believed that no computer realizable in our world can have capabilities surpassing those of a universal Turing machine. Though the definition of a universal Turing machine is quite simple—a machine places a read-write head over an individual square on a tape of unbound length and then moves the read-write head or writes a symbol, depending on the instructions associated with the symbol on the tape and the internal state of the machine—it was found to be theoretically capable of carrying out any determinate algorithm that a human mathematician can and equivalent to Alonzo Church's notion of effective calculability. Thus, work in theoretical computer science suggests that for practical purposes to be computable is to be computable by a universal Turing machine, and hence it is assumed that computation is best understood as activity of the sort doable by a universal Turing machine. Of course, the thesis that no physical device in our world can surpass the capabilities of a universal Turing machine remains unproven and, in all likelihood, is unprovable, since it is a conjecture about the physical nature of the empirical world.

One interesting observation regarding the question is one made by Richard Feynman. In "Simulating Physics with Computers," he noted that ideally, our world should be perfectly simulable by a classical computer and that "the number of computer elements required to simulate a large physical system [should be] proportional to the space-time volume of the physical system… If doubling the volume of space and time means I'll need an *exponentially* larger computer, I consider that against the rules (I make up the rules [of this thought experiment], I'm allowed to do that)."[3] However, he went on to note that classical computer appear to violate this rule and use exponentially more resources to simulate a quantum interaction than is proportionate. Accordingly, he proposed that physicists and computer scientists work around this apparent difficult

---

[3] Feynman, 134.

by performing simulations using other quantum objects in order to escape this exponential growth in resource requirements. While the issue of the relative power of quantum versus classical computing systems has not been resolved, the presence of this uncertainty suggests that we take seriously the possibility there are physical processes that seem to allow us to compute more than we "should" be able to compute using a given number of resources if we take the universal Turing machine as the be-all end-all of computing.[4]

Therefore, in order for one's definition of computation to be truly general, it must also accommodate the possibility of various forms of "hypercomputer" such as a computer that can perform calculations using the complete set of real numbers or a computer with access to an "oracle" function capable of predicting whether particular algorithms will halt. Indeed, if the world itself is a computer, then one possibility is that the seeming indeterminacy of quantum physics is merely a byproduct of the irrepresentability of quantum calculations in computers that are only Turing equivalent.

One happy exception to the general pattern of ignoring the need to distinguish between specific models of computation and computation in general is B. Jack Copeland's *The Broad Conception Of Computation*. In it, Copeland distinguishes between "Computable" with an uppercase 'C' and "computable" with a lowercase 'c.' The first refers specifically to computability by some Turing equivalent machine; the second refers to computability by any conceivable computing machine. Copeland describe the computing machine in general as follows:

---

[4] Note that there are two senses in which one computer can be more "powerful" than another. The first is one computer can, if given enough resources, solve a problem that another cannot, even if it is also given as many resources as it might want. In this sense, there is no question that a universal Turing machine is exactly as powerful as a quantum computer if both are allowed to have as much time and memory as needed. "Hypercomputer" is a term for computers more powerful in this first sense. The other sense of powerful is how many resources are needed to perform a certain family of calculations. It is in this sense that quantum computers appear to be more powerful than ordinary computers, although it has not been shown definitively.

> The general requirements for a computing machine are simple. Part of the machine must be capable of being prepared in configurations that represent the arguments of whatever functions are to be computed, and part of the machine must be capable of coming to assume, as a result of activity within the machine, configurations that represent values of these functions. Sub-devices of the machine—"black boxes"—must make available some number of primitive operations. The machine must have resources sufficient to enable these operations to be sequenced in some predetermined way. …[R]ecursive application, or iteration, of primitives is the essence of computation.[5]

Here we can see that Copeland has laid out a much broader notion of computation than what is generally considered. However, there are interpretative hazards when taking this definition to encompass all of computing. First, it can be criticized for it insistence on the inclusion of "black boxes" as sub-devices of the computer. The internal implementation details of the computer are of course abstracted away by the use of the term "black box," so this definition is not restricted to a specific kind of implementation of a computer as the definition of Turing machine is. However, even using the term "black box" at all strongly suggests the existence of internal (that is to say, systemically isolated) mechanical process by which values are calculated. A truly broad concept of computation cannot be so limited if it is to encompass the full range of possible non-digital computations, since unlike a black box, the functioning of some computers may not be so easily divided into internal components and external observers. Instead, the two may be dependent on one another, with information more readily available at their juncture, but with its arising dependent on the coordination of both the internal and external. Thus, the use of black boxes seems to suggest that the computer cannot use knowledge about the world as a whole as a part of the process of creating new information for the observer's consumption. Hence, by restricting computing to machines with black boxes, this concept seems to encompass only "complete" computers like universal Turing machines and the von Neumann architectures of contemporary personal computers and to exclude calculating tools like abacuses, in which the operation of the calculating device is partly dependent on the human operator even after the

---

[5] Copeland, 695.

initial programming phase of computation is complete. With the abacus, like a logarithmic table or even the lowly poster of the multiplication table common in elementary schools, a physical artifact is used to help make simple correlations between arithmetic entities. While an abacus may not seem to be "automatic" enough to count as a computer to some, I argue that the difference between the levels of operator involvement in the use of an abacus and an electronic calculator is one of degrees rather than kind. Just as you cannot learn the answer to two plus two using an electronic calculator without pressing 2, +, 2, =, and then interpreting the result on the screen, so too with an abacus, the operator cannot learn the answer to the same question without first sliding two beads, then sliding two more, and finally interpreting the resulting physical state of the abacus.

Further, the requirement of "primitive operations" being available for the computing machine seems to rule out the possibility of a physical artifact with an unbound number of possible state transitions. For example, it is not clear that a slide rule must in principle have only a finite number of primitive operations. Obviously, any actually existent slide rule will be constructed only within certain physical tolerances, and an adjustment either to the left or right a distance of less than that tolerance could be considered the primitive operation of such a slide rule, since motions less than what is allowed by its construction tolerances cannot be counted on to give accurate results. However, these limitations are practical ones, rather than theoretical ones. For an ideal slide rule, any motion—no matter how small—will result in a slight change in its output. In such a case, it is not clear that there are any truly primitive operations. Further, we might envision a computer in which the primitive operations of the computer change over time, as the machine "evolves" mechanically. In a science fiction scenario, an intelligent computer/robot may upgrade itself periodically, resulting in an ever shifting basis for its calculating substrata and unfixing the number of its primitive operations. More seriously, the precursors of today's internet have been in continuous operation since the late nineteen-seventies, during which time its basic protocols have been revised substantially. One can easily envision a network of computing devices which is also used without interruption to work on the same calculation for years, during which time its most fundamental computational substructures are completely replaced without halting the overarching activity of the network.

Finally, we note that caution should be taken when using the language of "computing machine," as it seems to suggest the necessity of human artifice for the construction of the machine, whereas if computation is possibly the fundament of the world, human artifice is ruled out *ex hypothesi*.[6] John Wheeler, a proponent of digital physics, proposes that we reject language of "machine" because it "has to postulate explicitly or implicitly, a supermachine… which will turn out universes in infinite variety and infinite number."[7] Accordingly, it is desirable that our definition of computation encompass the possibility of computing without human involvement. Of course, there remains the possibility that there is a Divine Programmer who understands the world program and in reference to Whom the universal computing machine is a machine, but in such a case, it is God rather than the program which is truly fundamental, and the paper is concerned with the possibility that *daode* is a computation-like fundament, not a God-like fundament or a tool used by God. If possible, a definition of computer that does not depend the anthropic perspective would be preferable. (We will explore the question of whether such a definition is possible in great depth later in this paper.)

In order to work around these interpretative difficulties with Copeland's definition of computing machine, the definition of computation that will be proposed here takes simulation to be the ground of all computation. In order to encompass as broad a range of possible computations as possible, we must not mistake numbers, discrete output, or algorithmic processes for what is mathematical in the most fundamental sense. As

---

[6] Excepting the scenario envisaged by Isaac Asimov's science fiction classic "The Last Question."

[7] A quote from Wheeler's "World as system self-synthesized by quantum networking," contained within his "The Search For Links" page 314.

Heidegger says in “Modern Science, Metaphysics, and Mathematics,” “the essence of the mathematical does not lie in number, as purely delimiting the pure ‘how much,’ but vice versa,”[8] since the mathematical is “the fundamental presupposition of the knowledge of things.”[9] Thus, as a presupposition of knowledge, what is fundamental to the concept of mathematical computation is not the ability to do arithmetic but an ability to predictively simulate some system in advance through the prior comprehension of the system. For example, in a wind tunnel, a model airplane is used to predict the behavior of a real airplane in flight. This device for making predictions connects one physical system to another without stopping to manipulate anything that “represents” numbers or other mathematical entities within its system, but the calculations that it does are no less deterministic and useful within their known tolerance levels for this.

Thus, we agree with the first part of Copeland’s proposed definition and take it that the key to any computer is its ability to “represent” various system states and their connection. To put it succinctly, computation is “**the process by which the fact that one system is rule governed is used to make reliable inferences about another rule governed system**.” This definition overcomes the interpretive difficulties of Copeland’s definition by emphasizing that computation is a systematization of analogs. Thus, in the case of the abacus, the fact that the beads won’t merge or divide spontaneously allows us to use them reliably for making finite determinations about the relationships of natural numbers. Similarly, even an idealized slide rule, an evolving computer network, or a wind tunnel can be seen as a rule governed system, even if those rules are not easily enumerated or even fully grasped by an observer of the computing system. Finally, this definition allows for any rule governed systems to be correlated, whether these systems are natural or artificial. Accordingly, the remainder of this essay will be conducted with this definition of computation in mind.

---

[8] Heidegger, 277.

[9] *Ibid*. 278.

III. Correlations between *daode* and computation

With our new definition in place, certain similarities between *daode* and the computer become clear. In both, the macrocosm is correlated with the microcosm. As a case study to illustrate this connection, in this section we will contrast the process of divining the weather using the traditional “oracle bones” method of ancient China with modern weather predicting computers. During the Shang and early Zhou dynasties, diviners would interpret the cracks that emerge on turtle shells and other bones on the theory that the state of the world as a whole would influence the patterns that emerged on the shell’s surface. When a proper ritual rubric is carried out, various inferences about the future can be reliably made. For example, one might ask whether the upcoming harvest season will be auspicious or inauspicious. Though the process by which these predictions were carried out appears completely dissimilar to modern methods of weather prediction, there is an underlying unity of form. In both processes, first proper preparations must be made in order for the process to work correctly. Then a complicated series of correlations are carried out by a physical substratum. Finally, the changes in the physical substratum are interpreted as an answer to the question posed. The apparent dissimilarity between these two activities is due to an understandable bias: the belief that oracle bone predictions do not work. If we bracket the concerns that we modern scholars may have about whether divination actually works, then we must say that if it does work as claimed, then the reality that allows it to work is a process of correlations that falls under our definition of computation. One rule governed system (the oracle bone divination equipment or the computer simulation) is used to make reliable inferences about another rule governed system (the world as a whole). The reason that both systems work is that the microcosm, be it of the computer chip or the diviner’s tools, is made to reflect the macrocosm, and once the correlation is firmly established a change in one can be read as a change in the other.

Of course, before accepting this analogy between a computer prediction and a divination, we need to reexamine each of the parts of the process—the preparation of the system, the change of the physical substratum, and the interpretation of result—before finally the system as a whole is examined and the overall relevancy of the connection between the two systems is assessed.

A. Preparation

In the preparation of the process, it may be objected that the preparation carried out in a computer simulation is the gathering of data, that is numerical facts, whereas the preparation process in divination consists primarily in the invocation of a question following a long ritual process of cleaning and polishing the bone to be heated. However, two points can be made regarding these dissimilarities. First, the reason that data is not gathered in a divination is that the gathering of data is assumed to be superfluous, since in traditional Chinese cosmography, everything is a correlate of everything else, so the way to learn about what is most distant is to study what is closest at hand. Thus, it is important that our definition of computation allows for the "black box" of the computer to be non-self-contained and interconnected with the system that surrounds it (hence not a "box" at all).

Secondly, that divining relies on the invocation of human language is not necessarily damning to its computational equivalence or proof that what is being interacted with is primarily a human-like and non-computational spirit. As David Keightley explains in *Sources of Shang History*, "the scapulimantic and plastromantic inscriptions were not simply regarded as prayers or magical letters forwarded to the spirits."[10] Rather, the inscriptions were used to track the accuracy of the divining process and thus to ensure that processes which produce more accurate results are retained and refined and those who master them are properly rewarded. While there was perhaps some anthropo-

---

[10] Keightley, 45.

morphic element to Shang era divination, by the time of the emergence of Daoism, *tian* 天 (often translated as "Heaven") had come to be seen as an impersonal force which orders the world.

Furthermore, we ourselves regularly use human language to interact computationally with the physical world. In the programming language Python, one makes the computer display the message, "Hello World!" by giving the input `print "Hello World!"`. Hence the difficulty with seeing ancient Chinese divination as computational is not that human language is used. The difficulty is whether it is plausible that their language can really be transformed reliably into other physical processes that in turn can correlate properly with the world as a whole.

Of course, if one is a computationalist about the world as a whole, one is likely to be a computationalist about human consciousness as well, in which case, if human language is meaningful in any sense whatsoever, it must directly relatable with other physically instantiated computational structures, i.e. the ones in our brains, which leaves open the possibility that the "spirits" (if real) are also underwritten by a computational substructure. In any event, it is not *prima facie* implausible that there can be a reliable analogs between human language and the physical world.

B. Physical change

The change of the physical substratum presents another area for critical consideration. In a digital computer, the electronic signals of data recording devices are made to correlate with electromagnetic fields in other calculating devices, then the pattern of these gives rise necessarily and mechanically to new signal patterns which propagate and interact before finally leading to their expression as an image on a monitor or a sheet of paper. In contrast, the changes in the physical substratum of a divination seem more straightforward. Heat causes the material of the bone to expand until it cracks. However, if we ask why the bone cracks where it cracks, then the process of explanation is much more involved. The bones used for divination represent a “found” computation. It is not a product of human artifice that bones used will crack when heated properly. However in a Chinese cosmology that the bones do crack precisely where they do is a matter of their interrelation in the cosmology of *yin* 陰 and *yang* 陽 leading to a particular physical outcome. Of course, this interrelation is not merely a matter of simple causality in our contemporary sense. As A.C. Graham explains in *Disputers of the Dao*, causal thinking was not present in the sciences of ancient China. For this reason, as he explains,

> It is not just that the explanations of Chinese as of Western Medieval and Renaissance proto-science may impress us as obscure or fallacious like the arguments of the philosophers; the trouble is that for post-Galilean science they are not explanations at all.[11]

As correlative rather than causal systems, the cosmologists of these sytems did not emphasize the ways in which certain isolated states of matter necessarily give rise to one another, as contemporary science does. As Graham states, the world of Chinese proto-science is, “a world is which not, like that of Newton, bound by invariable law.”[12] Nevertheless, ancient Chinese cosmology is still at least an nondeterministically rule governed system along the lines of a quantum system, rather than a truly lawless anarchy. As Graham goes on to explain about the Five Processes (*wu xing* 五行) of Chinese cosmology,

> It was noticed from an early period that the processes conquer each other in a regular cycle, water quenching fire, fire melting metal, metal cutting wood, wood digging soil, and coming round again with soil damming water…[13]

The *Daodejing* is also presupposed on the indeterministic but rule governed nature of the cosmos. As chapter 16 explains in part,

> In the process of all things emerging together (*wanwu* [萬物])
> We can witness their reversion.
> Things proliferate, And each again returns to its roots.[14]

---

[11] Graham, 320.
[12] *Ibid*. 354.
[13] *Ibid*. 326.
[14] Hall and Ames (2003), 99.

Thus, all change is based around a common center point from which things reliably emerge and to which things reliably return. Furthermore, it is clear that in the processual nature of *daode*, physical transformations that take place on one level of reality always reflect a micro/macrocosmic recapitulation these similar transformations at every other level of reality. As the end of chapter 25 of the *Daodejing* states,

> Human beings emulate the earth,
> The earth emulates the heavens [天],
> The heavens emulate way-making [*dao* 道],
> And way-making emulates what is spontaneously so (*ziran* [自然]).[15]

The character translated as "emulates" is 法 (modern Mandarin *fa*), a term that would later by taken by the Buddhists to translate dharma and was also emphasized by the Standardist school. In modern Chinese and Japanese, this same character is used to translate the Western concept of a law. What its translation here as "emulation" correctly emphasizes is that the changes that arise in this world are rule governed in such a way that each level of activity can be correlated with each other level, and that in order to gain greater insight into one particular level, it is often pragmatically useful to study some other level that is closer at hand instead of looking directly at the level towards which an inquiry is posed. This, of course, falls directly under our working definition of computation.

C. Interpretation

Concerning the issue of interpreting the results of the computation or divination, A.C. Graham seems to suggest that divination in the *Yijing* 易經 (and by extension oracle bone divination) does not really work according to the mechanism accepted by its practitioners, but rather it works by allowing for the tension between the heuristic of the divination and the reality of the situation perceived by the diviner to be worked out through creative interpretation.[16] Thus, divination is an aid to creative thought and

---

[15] *Ibid*. 115.
[16] Graham, 368.

not an direct correlate of the physical world. At the opposite extreme, François Jullien in *The Propensity of Things* can be interpreted as ascribing to Chinese thinkers a belief that *shi* (勢, a term for "momentum" that seems to bridge *dao* and *de*) has a correlation with the actual world strong enough that "they could detect in warfare's unfolding a purely internal necessity that could be logically foreseen and, accordingly, perfectly managed."[17] Between these two views of the actuality of the connection between Daoist processes and the world, we must note that even an ordinary computer simulation must be itself interpreted. The number on a screen that results after the computer has finished its calculation means nothing in itself, but rather, it takes on meaning only when interpreted in light of the mental worlds of the programmer and the user. Although Jullien notes that *shi* inserts itself, "into the distinction between what Westerners have opposed as 'practice' and 'theory,'"[18] we can say the same about computation as well. Computation takes the most theoretical of disciplines—mathematics—and drags it down in the practicality of observable physical transformations.

D. Process as a whole

Finally, let us consider a computer program as a whole. If we have the text of a program like the aforementioned `print "Hello World!"`, by itself we have nothing, since a program cannot be run without a system. Furthermore, in one important sense, the text of the program is not the program itself. Neither is even the machine code (the "ones and zeros") to which the text program is converted the program itself. The program as executed is a real collection of electrons and magnetic fields here and a real absence of electrons and magnetic fields there. As such, the same program in the sense of its fullest actuality is never seen twice, since the electrons in a CPU are a veritable Heraclitean river of flux. Worse still, according to quantum physics, the electrons themselves are not

---

[17] Jullien, 25. Note that Jullien's own view of the role of *shi* does not seem to be this mechanized. However, since his language allows for this interpretation, we will use it here as a convenient foil.

[18] *Ibid*. 38.

static entities, but clouds of potential. The existence of a program as we think of it is an abstraction from these actual clouds of potential—and thus in one sense "unreal." When we define a computation as the reliable inference about one rule governed system from another, the system that is being inferred about using the physical system is a logical system where the text of the program is true. Thus, in computing the program, we bring it out of the theoretical by tying it with some actual—the actual physical make up of the computer running the program. What Hall and Ames say of the *dao*, we can also say of the relation of the text of the program to the physical substratum of the program: it is merely a convenient collection of "thises and thats" that allow us to make meaningful abstractions about the real world. The text of the program allows what is beyond our comprehension (the swirling mass of uncounted swarms of subatomic particles) be made within our comprehension (the program text) in service of finding some result that we have not yet brought out from our comprehension into reality (the result of running the program).

The same relationship that holds between a program and a computation can be said to hold within *daode*. Hall and Ames give a "bare-bones reading" of *Daodejing* chapter 42 thus,

> *Dao* engenders one,
> One [engenders] two,
> Two three,
> And three, the myriad things.[19]

They also give a fuller translation,

> Way-making (*dao*) gives rise to continuity,
> Continuity gives rise to difference,
> Difference gives rise to plurality,
> and plurality gives rise to the manifold of everything that is happening (*wanwu*) [20]

---

[19] Hall and Ames (2003), 143.
[20] *Ibid*. 142.

We can interpret the passage in the following way. *Dao* is the condition necessary for transformation to occur: the system as a whole. As the *Yijing* says,

> Thus, that which goes beyond form is called *dao*; those things that have form are called phenomena. The transforming and tailoring of things is called flux. The extending and applying of things is called continuity. To take up this understanding and bring it into the lives of the common people is called the grand undertaking.[21]

Within the system when a particular foci or *de* arises, this naturally begins a process of change and counter-change that results in the complete structuring of the ten-thousand things. Under this interpretation, *dao* is the computer; *de* is a program; the generation of the ten thousand things happens as a result of their iterative interaction. For humans to understand the world, we must understand the system that gives rise to the world, then apply this understanding to specific cases through the grasping of their particular foci, then turn that understanding to our own advantage as we use the iterative nature of the world to our own advantage. As chapter 64 of the *Daodejing* states, we must "Deal with a situation before it happens."[22]. Our ability to do so is in turn dependent on our understanding of *daode* giving rise to actions that best conform to the eventual happenstances of the evolution of *daode*.

IV. Differences between *daode* and computation

Having made the case for interpreting *daode* as computation, we must now refine our interpretation by noting the areas in which our previous interpretation fails to properly characterize Daoist thinking. There are several categories into which these failures fall. First, there are problems with attempting to give a single characterization to the *dao*. Next, there is the importance of *de* to realizing *daode*. Finally, there are general problems with taking computation as primordial.

A. Difficulties with characterizing *dao*

To begin, let us turn to the very first words of the *Daodejing*,

---

[21] Translation from *Yijing*, Xici 1, 12 courtesy of Roger Ames.
[22] Hall and Ames (2003), 177.

> Waying-making (*dao*) that can be put into words is not really way-making,
> And naming (*ming* [名]) that can assign fixed reference to things is not really naming.[23]

With these words in mind, it is clear that any concept of *daode* as computation will run into serious difficulties if it naïvely assumes that *dao*, like a computer program, can be given a single, fixed explanation. We have already argued against the conception of *daode* as either a simple, causally determinate system or an absolute chaos. As chapter 34 of the *Daodejing* states, "Way making (*dao*) is an easy-flowing stream/ Which can run in any direction."[24] Literally, the passage states that the *dao* "can go left and right." This strongly suggests that the spontaneity of *daode* cannot be pre-emptively circumscribed by human beings. The difficulty then is in determining to what degree *daode* can be considered to be rule-governed, when the rules that govern the unfolding of *daode* are not extrinsic to it, but intrinsic yet unfixed.

Simple indeterminism is not enough to dissuade us from a computational interpretation. Chaitin's constant Omega, for example, is a mathematical entity that is essentially random, in the sense that it cannot be understood in terms of anything simpler than itself, yet it can be formally defined in a relatively simple manner as the halting probability of any algorithm. Thus, by building on the work of Turing and Gödel, Chaitin has demonstrated a "random" number that can be constructed within the rule governed system of mathematics. The trick of Chaitin's constant and related constructs is that they are only random in the sense that their value is externally inexplicable. It is perfectly determined internally, but working out precisely what it is determined by the simple definition of the constant is an intractable problem that would, in principle, require more than an infinite amount of conventional computations effort because of its incredible information density.

---

[23] *Ibid*. 77.
[24] *Ibid*. 130.

Whether *daode* follows this pattern then depends upon whether *daode* is epistemically or ontologically ineffable. Algorithmic randomness relies on epistemic ineffability to conceal its ontological determinacy. Whether *daode* also relies on epistemic limitations in order to conceal ontological unity cuts at the heart of a central question of contemporary Daoist studies: Is *daode* transcendent or emergent? If *daode* is in fact transcendent, then there is in fact a constant *dao* that exists independently of our knowing. If *daode* is emergent, then *dao* and *de* are self-arising and without an external source of structuring. This paper cannot hope to fully settle the question of the transcendence or emergence of *daode* within the limits available here, but some remarks will be made.

In *Thinking from the Han*, Hall and Ames define strict transcendence as "*A* is transcendent with respect to *B* if the existence, meaning, or import of *B* cannot be fully accounted for without recourse to *A*, but the reverse is not true."[25] In the case of computation, transcendence is encountered whenever one of the rule governed systems being related exceeds in its actuality the representational capacities of the other system. Thus, for an ordinary computation, we can say that the physical system transcends the computational interpretation imposed upon it, since there is a loss of information as one goes from the messy reality to the smoothed over interpretation of that reality. (That is, there are always a myriad of physical states that have the same computational effect, since, as previously mentioned, while we talk about the current in the wire not being a "one" or "zero," in reality, there are small, insignificant fluctuations in voltage that are covered over by the interaction of logic gates, and are thus unknowable on the basis of the output of the computation alone.) In a computational world, this order of transcendence would be reversed as the physical world would ultimately depend on the epistemically inaccessible Great Computer. Like the Kantian noumena or the physicist's Theory of Everything, the Great Computer as the reality behind the illusion would only be

---

[25] Hall and Ames (1998), 190.

the subject of speculation, not knowledge, for us physical beings. If the Dao is like the Great Computer, then it too should transcend the physical world. However, can we find support for this in traditional Daoist texts ?

Perhaps not. Chapter 62 of the *Daodejing* ("Way making (*dao*) is the flowing together of all things (*wanwu*)."[26]) strongly suggests the emergence rather than transcendence of *daode*. While we earlier explained how chapter 42 relates the emergence of all things from *dao* as the one gives birth to the two, etc., nevertheless, other evidence suggests that this process should not been seen as statically unidirectional. It is just as proper to say that the myriad things give birth to the three, which gives birth to the two, etc. The reason for the viability of this reversible comprehension is that the origin of the world is not seen as one time event, but an on-going process contained in each moment.

An example of this atemporal structure is seen in *Daodejing* chapter 40, in which it is explained that,

> "Returning" is how way-making (*dao*) moves,
> And "weakening" is how it functions.
> The events of the world arise from the determinate (*you* [有]),
> And the determinate arises from the indeterminate (*wu* [無]).[27]

The last two lines might also be rendered as, "Things under heaven arise from what there is, and what there is arises from what there is not." Thus, neither being nor non-being is primordial in the working out of *dao*, and neither rule governed determinism nor chaotic indeterminism is more basic to its nature. Rather, the *dao* works by undermining such oppositions, erasing their incommensurability in its "returning" and "weakening."

A concrete illustration of this reversibility can be seen in *The Great One Gives Birth to the Waters*, a text that was apparently once a part of the *Daodejing*, only to be lost until it was found again in an archeological dig in 1993.[28] *The Great One* begins by explaining

---

[26] Hall and Ames (2003), 173.
[27] *Ibid*. 40.
[28] *Ibid*. 225.

how the Great One gave birth to the heavens in collaboration with the waters, which in turn leads to other births culminating in the creation of the yearly cycle. Having explained all this, the text then traces its way backwards from the yearly cycle back up to the Great One.[29] The purpose of this reversal is to show that while there is a unidirectional order of birth, with the Great One at the top and the yearly cycle at the bottom, there is not a unidirectional order of priority. While the year cannot be said to give birth to the Great One, nevertheless, the Great One would not be the Great One without the existence of the yearly cycle. Thus, the Great One does not transcend its descendants, nor is it transcended by them.

If *daode* is similarly non-transcendent, then it cannot be perfectly simulated from without, because any external simulation would need to rely on the existence of complete structural similarities between the calculating system and *daode* in order to create a system of reliable inferences between the two systems, and such complete mirroring of structure cannot exist if mirror needs to also mirror itself. Thus, the mysteriousness of *daode* arises not from its distance from human life, but its closeness to it. As chapter 38 of the *Daodejing* states, "'Foreknowledge' is tinsel decorating the way [*dao* 道],/ And is the first sign of ignorance."[30] That is, thinking that we know the way before we actually do is the first step on a false path. Truly grasping the way means being aware of its ineffability and resolutely able to accept whatever the way may bring.

Thus, the first major blow to the computational interpretation of *daode* is that, unlike a computer system, while *daode* is rule governed, it neither transcends its effects nor is it transcended by them.

B. Difficulties accounting for the role of *de*

One reason that we first had such facility at analogizing *daode* to computation, only to later encounter difficulties with computational transcendence is that so far our exposi-

---

[29] *Ibid*. 229 – 230.

[30] *Ibid*. 136.

tion has accounted for *dao* to the near exclusion of *de*. Loosely, one can say that if *dao* is the "potential" of the system, then *de* is "actuality" of it. (Of course, this is only loose talk, since both *dao* and *de* contain measures of each.) In the focus-field model of Hall and Ames, *de* is the focus in which the "insistent particularity" of the whole comes to the fore. *De* are those concrete realities which keep *daode* from becoming a swamp of abstract generalities, but at the same time, *de* and *dao* are structured in such a way that each is contained within the other. Within the *Daodejing*, *de* is repeatedly stressed as the key to proper understanding of how to make one's way in the world. For example, chapter 65 explains that in the area of statecraft,

> Those who really know the distinction between using knowledge or a lack of it in governing the state will moreover become its model.
> And those who really know this model are said to be profoundly efficacious (*de* [德])[31]

Thus, *de* is the profound efficacy that comes from knowing one's limitations. Along the same lines, chapter 38 extols the effortless success that comes from understanding particulars:

> It is because the most excellent (*de* [德]) do not strive to excel (*de*)
> That they are of the highest efficacy (*de*).
> And it is because the least excellent do not leave off striving to excel
> That they have no efficacy.
> Persons of the highest efficacy neither do things coercively
> Nor would they have any motivation for doing so.
> … Thus, only when we have lost sight of way making (*dao*) is there excellence[32]

Hence, again we see that true excellence comes from an understanding that does not seek to control or rationalize—a major theme of the *Daodejing* that is difficult to account for in the computational model of *daode*. Time and again, the *Daodejing* points its readers to the importance of what is not (無), the obscure (玄), the mysterious (妙), and the feminine (牝). A proper understanding of why these aspects of reality are so emphasized begins with an understanding of the importance of *de* in *daode*.

---

[31] *Ibid*. 179.
[32] *Ibid*. 135 – 136.

The alternative to a Daoist strategy of non-coercively coping with ignorance is one almost invariably taken in the software development field—planning out in advance every eventuality and pre-programming a response for each occasion. For example, one famous problem for conventional, contemporary computers is that attempting to divide by zero may cause a software crash. For this reason, conscientious developers need to study every part of their code where two numbers may be divided, think about the possible inputs at that stage of the code, devise an appropriate response to a request to divide by zero, and insert a test that ensures that an appropriate response is carried out in the event of a zero division. Of course, division by zero is only one of the best known of a large number of potentially destabilizing software operations. There are innumerable others, such as buffer overflows, improperly terminated strings, memory leaks, and so on. Worse still, even minor hardware malfunctions, such as a bit being changed by cosmic radiation, may cause a cascade of further errors if the malfunction occurs in a critical juncture. In order to avoid these problems, the developer must use their foresight at every step of the way to identify any and all potentially problematic areas and write an appropriate response. Unfortunately, finding these "corner cases" through routine planning or testing can be very difficult, since it is precisely the unanticipated nature of these bugs which makes them so dangerous. Of course, it is easy to make a computer program that has a response to all situations. One could, for example, program one's computer to always resume operation after a division by zero as though no request to divide had been made. Unfortunately, such a response would not be appropriate in the majority of cases and would lead to undesired results, which is the reason that it is not already the procedure followed in the event of a division by zero. While it is easy to have a response for any situation, it is much more difficult to have an appropriate response for all situations. The measurement of appropriateness of response is too fine a calculation to make in general on the basis of anything less than everything all together.

Computation is not the only field where the difficulty of crafting appropriate response rears its head. To give just one more brief example, in the field of law, it is normal for a law to be written only after an offense has occurred once and it is recognized as something to be prevented. Since even simple laws may be worded in such a way as to lead to unexpected consequences, the law must be revised periodically in order to deal with an unforeseen situation that arises. The difficulty for both the computer software and the law is that programmatic rules exist to smooth over differences between situations and create a uniform protocol of response. Unfortunately, in the real world, there are too many unexpected occurrences and implications to create an abstraction that can smooth over all distinctions without also losing fine details. These lost details have a habit of accumulating over time and leading to crises. The Daoist emphasis on the *de* of *daode* is precisely a reminder of the dangers and opportunities that arise out of the accumulation of lost details. For the Daoist sage, these unexpected *de* occurrences are not a problem to be either avoided or mastered in advance through careful pre-programming of responses, but rather, it is the emergence of these occurrences which is to be encouraged and coped with as they arise. They are the source of novelty and originality in *daode* and what separate the spontaneous yet appropriate flow of *daode* from the inexorable grinding of a single, mechanistic computation on the one hand and the sheer randomness of cosmic dice on the other.

Among the qualities lost when one focuses on *dao* to the exclusion of *de* is human creativity. The point of mastery of *dao* is the expansion of human creativity to such a degree that even formidable, unforeseenable situations can be easily coped with. Pablo Picasso is credited (most likely erroneously) with the aphorism, "Computers are useless. They can only give you answers." This precisely captures the difference between Daoist thinking and computational thinking. In order to treat a system as a reliable means of correlating with another system, it is necessary to abstract away the irregularities of each system in order to emphasize the reliability of their connection. Modern computers are binary, because early computer scientists had too many difficulties creating reliable circuits that use more than two different voltages internally. This is just one example of how in order to understand the systems correlated, it is first necessary to remove extraneous details. However, the *Daodejing* repeatedly warns of us of the dangers that come from this willful ignorance of the seemingly insignificant. As chapter 71 informs us, "Knowing that one does not know is knowing at its best."[33] In order to know our own limitations, it is necessary to emphasize the mysterious and hidden so that we are challenged to ask new questions and create new, uncalculated responses. These responses optimize the situation without seeking to dominate it. They gain awareness without becoming limited by the narrow scope of their self-supposed knowledge.

C. Difficulties with the ontological primordiality of any kind of computation

In addition to the difficulties already listed with supposing that *daode* can be conceived of as computational, there are also reasons to question whether the world itself could be computational either. We earlier defined computation as "the process by which … is used to …." Here, the passive voice has elided an important distinction. Put into the active voice, the definition becomes, "the process by which we, conscious observers, use … to …." In other words, what is computation and what is not computation is a matter of subjective inference.

This in turn creates two possible definitions for "computer." A computer may be anything that *could* be used for computation. In this case, everything physical is a computer, since if nothing else, it could be used to assist in counting by assigning it a number. On the other hand, a computer may be anything that *is* being used for computation. In this case, a new laptop still in the box is not yet a computer, since its potential for use in computation has not yet become actual. Applying this to the cosmos as a whole, we find that either the cosmos must be a computer, no matter what its underlying ontolo-

---

[33] *Ibid*. 189.

gical makeup may be, or the cosmos was not a computer until conscious observers came into being, but now it is. Neither result is particularly appealing or enlightening, and both point to the unresolved difficult first hinted at when we began to define computation. Though it is possible to create a definition of computation that removes the implication of human artifice in the creation of the computer, it is not possible to create a meaningful definition of computation that omits the role of the human observer

Regarding the idea that computation can exist as computation independently of its recognition by an observer, Jaron Lanier proposes an interesting thought experiment that shows the danger of accepting this premise along with the computational theory of mind:

> Some people might remember the "rain drops" argument. Sometimes it was a hailstorm, actually. The notion was to start with one of Daniel C. Dennett's thought experiments, where you replace all of your neurons one by one with software components until there are no neurons left to convert. At the end you have a computer program that has your whole brain recorded, and that's supposed to be the equivalent of you. Then, I proposed, why don't we just measure the trajectories of all of the rain drops in a rain storm, using some wonderful laser technology, and fill up a data base until we have as much data as it took to represent your brain. Then, conjure a gargantuan electronics shopping mall that has on hand every possible microprocessor up to some large number of gates. You start searching through them until you find all the chips that happen to accept the rain drop data as a legal running program of one sort or another. Then you go through all the chips which match up with the raindrop data as a program and look at the programs they run until you find one that just happens to be equivalent to the program that was derived from your brain. Have I made the raindrops conscious? That was my counter thought experiment. Both thought experiments relied on absurd excesses of scale. The chip store would be too large to fit in the universe and the brain would have taken a cosmologically long time to break down. The point I was trying to get across was that there's an epistemological problem.[34]

While the imposition of panpsychism may not be unappealing to all scholars, nevertheless, the danger that Lanier's thought experiment appears to demonstrate is that unmoored from an actually operational frame of reference, anything can be made to represent anything else. Thus, we cannot distinguish between computational and non-computational entities on an objective basis. To say something is "a computer" is akin to saying something is "useful." Any object can be useful given the right agent in the

---

[34] Lanier.

right circumstances. Things may be commonly construed by us as "useful objects," but this is merely a shorthand way for us to say that we believe that is more likely that these objects will get a chance to manifest their potential for usefulness than other objects. So too, calling something a computer means merely that we believe it to be more likely than usual that object so described will be used for making truly complicated computations.

If then being a computer is not an objective aspect of entities in the world, then the notion that the world itself is at its most basic level a computer must be rejected if our subjectivities are presumed to rest on the objectivity of the outer world. At best, we can say that viewed from a certain perspective, the rules that govern the behavior of particles may be similar to recursively executed computer algorithms, à la Wolfram, but even if that is the case, it would not be strictly correct to call the world a computer, since "being a computer" is something that depends on the construing of an observer. There are then two possible ontologies to work out on this basis. In the first, the fundamental parts of our world are what they are, but they are most easily described by computational formulae. Such an ontology is only different in specifics from the ordinary scientific materialist ontology. In the other possible ontology, the world is a true computation, but unlike other computations, there is no physical basis for the computation which produces the physical world. Rather, a particular transcendent subjective perspective on the world gives rise to our own subjectivities through its iterative development. This ontology seems to be no more than a newly refined sub-category of theism, rather than a radical new ontology as computationalism promised to usher in. In either case, the fact of computationalism is not fundamental, but at one remove from what is most basic. All of this further separates the computationalist perspective from that of Daoism, in which *dao* and *de* are at once fundamental constituents of the cosmos, and at the same time emergent members of it.

V. Conclusion

In the course of this paper, we have shown that while Daoism shares many attributes in common with the computational view, in the end, there remain significant difference between them. Specifically, the computational view of the world fails to account for the true comprehensiveness and thus emergence of *dao* and the insistent particularity and thus mysteriousness of *de*. Nevertheless, recognizing the shared background of processual analogizing shared between the two outlooks enables us to deepen our understanding of both, and further, it allows us to recognize the degree to which computational thinking pervades our ordinary thought already at this early juncture in the information revolution. Ultimately, however, the computational view is a merely subjective imposition on the world, and, unlike *daode*, it is not capable of sustaining an ontology in itself. By bolstering our view of computation with insights from Daoism, we hope to see future advancements in our understanding of both.

VI. Bibliography

Asimov, Isaac (1990). "The Last Question" in *Isaac Asimov: The Complete Stories, Vol. 1*. Broadway.

Copeland, B. Jack (1997). "The Broad Conception of Computation." *American Behavioral Scientist*, May 1997; vol. 40: pp. 690-716.

Chaitin, Gregory (2005). *Meta Math!*. New York: Vintage Books.

Feynman, Richard (1999). "Simulating Physics with Computers" in *Feynman and computation: exploring the limits of computers book contents*. Ed. Anthony Hey. Perseus Books, Cambridge, MA, 1999: pp 133 – 153.

Graham, A.C. (1989). *Disputers of the Dao*. La Salle, IL: Open Court.

Hall, David L. and Roger T. Ames (2003). *Daodejing, "Making This Life Significant": A Philosophical Translation*. New York: Ballantine Books.

——— (1998). *Thinking from the Han: Self, Truth, and Transcendence in Chinese and Western Culture*. Albany: SUNY.

Heidegger, Martin (1993). "Modern Science, Metaphysics, and Mathematics" from *What is a Thing?* in *Basic Writings,* revised and expanded edition. Trans. David Krell. Harper San Francisco.

Jullien, François (1995). *The Propensity of Things: Toward A History of Efficacy in China*. Trans. Janet Lloyd. New York: Urzone.

Keightley, David (1978). *Sources of Shang History: The Oracle-Bone Inscriptions of Bronze Age China*. Berkley: University of California.

Lanier, Jaron (2003). *Why Gordian Software Has Convinced Me To Believe In The Reality Of Cats And Apples: A Talk With Jaron Lanier*. Interview with the Edge Foundation Nov. 19, 2003. Available at `http://www.edge.org/3rd_culture/lanier03/lanier_index.html`. Accessed Nov. 18, 2007.

Turing, Alan (1936). "On computable numbers, with an application to the *Entscheidungsproblem*." *Proceedings of the London Mathematical Society*, 1936; S.2, 42: pp 230 – 265.

Wheeler, John A. (1999). "Information, Physics, Quantum: The Search for Links" in *Feynman and computation: exploring the limits of computers book contents*. Ed. Anthony Hey. Perseus Books, Cambridge, MA, 1999: pp 309 – 336.

Wolfram, Stephen (2002). *A New Kind of Science*. Champaign: Wolfram Media.

Zuse, Konrad (1970). *Calculating Space*. MIT Technical Translation.