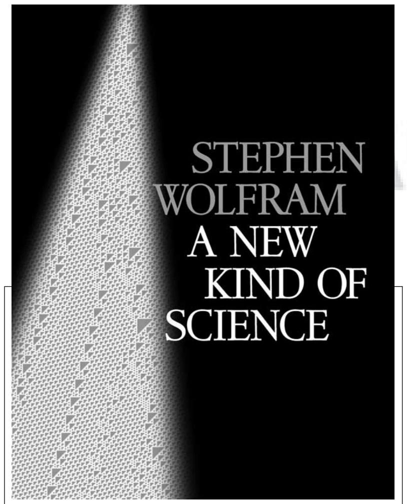
tapraid5/a5-cplx/a5-cplx/a50602/a50030d02o | deangeln | S=5 | 1/17/03 | 22:00 | Art: BR-84 | Input-DCT-msh

Mathematica's First Academic Monograph

F1 If the computer, HAL, from Arthur C. Clark's *2001: A Space Odyssey* were to write an academic monograph, you can be sure of two things. (1) It would possess many novel and important ideas because HAL is brilliant. And (2) it would be perplexingly odd because HAL's writing style would violate countless social conventions implicitly followed by human writers (the social conventions being particularly difficult for HAL to learn). Although HAL is fictional, history was made this year—just one year after the fantasy time period for HAL—when Stephen Wolfram rigged up his *Mathematica* software to write the first academic monograph ever generated by a computer program. This 1200 page book, titled *A New Kind of Science*, is indeed what one would expect from HAL: many new and significant ideas, and a chillingly alien, nonhuman presentation style and methodology that gives the reader a unique perspective into what it might be like to *be* an artificial intelligence. All ribbing aside, it really does *seem* as if something like *Mathematica*, not Wolfram, might have written the book: the repetitiveness, the immodesty, the title, beginning nearly every sentence with "and" or "but," the shortness of the paragraphs, his uncanny ability to refer to figures without numbering them, the utter lack of citations, the frequent use of the phrases "my suspicion is that…" and the quite distinct "my strong suspicion is that…," the grandiosity of his conclusions, the fact that there is more text in the endnotes than the main text (350 pages of double column small print), and, most of all, his ability to make all of these quirks seem as if they follow inexorably from some kind of internal book-writing logic. Despite all this, the book is eminently readable and serves as one of the best examples I have ever seen for how complicated ideas can be explained using just prose and (gobs of gorgeous) pictures. We could all learn a thing or two about presentation from his example. As for his thumbing his nose at academic etiquette—for example, by his lack of citations and references—it seems to me that when you are independently wealthy and self-made, thumbing your nose is your prerogative.



A NEW KIND OF SCIENCE
by Stephen Wolfram,
Wolfram Media Inc., Champaign, IL, 2002
1192 pp, $44.95 (US hardcover)

Let's put aside the *way* Wolfram says what he says. What *does* he actually say? There are more than a dozen paper's worth of work presented in the book. I will focus here on Wolfram's central thesis, his *Principle of Computational Equivalence*. Some preliminaries are needed. Imagine an infinite, linear array of cells all side by side, and suppose that each cell can be either black or white. Now suppose that the color of cells can change through time, and that how any given cell changes its color at the next time step depends only on the color of itself and its two adjacent neighbors. The fact that a cell's color at the next time step depends only on the nearby cells is important, for it makes cellular automata a nice way of modeling physical mechanisms of all kinds, and is one of the main reasons cellular automata are interesting in the first place.

The supposition above delimits a class of 256 particularly simple cellular automata rules, which can be classified into four broad types, depending on their behavior (i.e., depending on the manner in which the pattern of cell colors change through time). Type I rules are those that lead only to simple, repetitive behaviors. Type II rules are those that lead to order, nestedness, and certain fractal-like qualities in the behaviors. Type III rules are those whose behaviors appear to be random, with no long-range correlations at all, and Type IV rules are the interesting rules, those that appear to be sufficiently complex that no pattern can be seen in the behaviors, and yet there do appear to be long-range correlations. "Rule 110" is one of these Type IV rules and is the main character in the book. This rule just says the following: take "the new color of a cell to be black in every case except when the previous colors of the cell and its two neighbors were all the same, or when the left neighbor was black and the cell and its right neighbor were both white" (p. 31).

Now to the central mathematical result of the book, which concerns the notion of *computational universality.* A mechanism, or rule, is said to be computationally universal if for each computable function there is some initial condition such that the rule, when acting on that initial condition, is able to compute that function. (In effect, then, the initial condition is a program that computes the function.) As an example, all modern computers are computationally universal, in that for each computable function it is possible to put in inputs (an "initial condition") to your computer that make your computer compute that function, which is just to say that you can buy software (a purchased "initial condition") whose job it is to compute that function.

For cellular automata, the initial condition is the initial colors of all the cells in the linear array, and in the 1970s and 80s it was shown that there were computationally universal cellular automata with nearest-neighbor rules (see Wolfram's endnote on p. 1115). These were important results, for it meant that nearest-neighbor physical-mechanism-like rules can do anything a computer can do. And these mechanism-like rules are inherently more interesting to scientists wishing to model the world's mechanisms; e.g., in biology, a computer is not a good way of thinking about how groups of cells interact, but a cellular automaton is. The computational universality of cellular automata opens up the possibility for computational universality existing in the world (e.g., in groups of cells). But these early computationally universal cellular automata rules were very complicated (requiring lots of colors, not just black and white), leaving one to doubt whether computational universality could really ever be expected to occur in nature.

Wolfram suspected that Rule 110, which is *much* simpler than the previously known computationally universal cellular automata rules, was computationally universal, and—here comes the central mathematical result of the book—a student/assistant of his named Matthew Cook was able to prove it. That is, although Rule 110 is a simple rule, it is sufficiently powerful that, for any computable function, there exists an initial condition of cell colors such that the evolution of the cells amounts to a computation of the function. If computational universality can not only occur for mechanism-like processes, but *simple* mechanism-like processes, then perhaps computational universality actually occurs in nature!

Which brings us to the Principle of Computational Equivalence. The principle has two parts. The first is that there is a limit to computational power: nothing in the universe can compute anything that is not Turing computable (i.e., computable given a sufficiently large computer). Except for those like Penrose [1] who argue that brains can compute the Turing-*un*-computable, I suspect few will find this controversial. At any rate, this "computational limit" half of the principle is not particularly unique to Wolfram (e.g., see Changizi [2]). What is new is the second half, which concerns the commonness of computational universality: "almost all processes that are not obviously simple can be viewed as computations of equivalent sophistication" (p. 717), and, in particular, "should thus be in effect [computationally] universal" (p. 718). Despite unambiguous writing throughout most of the book, here Wolfram becomes cryptic, and it is accordingly difficult to pin down what he means by this. The best interpretation I have been able to muster is that *almost all processes or behaviors that are not obviously simple are due to an underlying rule or mechanism that is computationally universal.*

But why should complex-seeming behavior lead us to expect that the underlying mechanism is a computationally universal one? To help us answer this, consider behavior now in terms of the activities of a Turing machine when run on some input. (A Turing machine can be understood, for our purposes, as a set of rules directing a Turing machine head to move back and forth along a tape, reading and writing. The input to the Turing machine is written on the tape, and the *behavior* of the Turing machine on that input is the pattern of activities carried out by the head.) It suffices to look at these Turing machine behaviors, rather than cellular automaton behaviors, because of their computational equivalence (see above). (It is also where I am more comfortable thinking.) Suppose that a Turing machine's behavior B when implementing algorithm A on input x is not obviously simple. What can we conclude about the rules governing this particular Turing machine? That is, what can we conclude about *which* Turing machine is probably generating behavior B via algorithm A on input x? There are broadly two kinds of Turing machine that could be underlying the behavior. Possibility 1 is that the Turing machine is *not* a universal Turing machine (i.e., not computationally universal), but is merely a machine that implements algorithm A; it is an algorithm-A-Turing-machine, i.e., a machine "hard-wired" to carry out just that algorithm. Possibility 2 is that the Turing machine *is* universal, and happens to have "loaded" on its tape software for computing algorithm A. Either way, the same algorithm is implemented, and the same behavior B is exhibited when *x* is the input. (In fact, the same is true even if, by "behavior," we refer the set of all behaviors exhibited by algorithm A over all inputs.) So, which Turing machine probably underlies behavior B? And why should B's being not obviously simple make Possibility 2 more probable? Wolfram provides no answer. One conceivable answer sketch might be that, for biological processes and behaviors, once evolution stumbles upon a computationally universal mechanism, organisms employing it tend to take over (for reasons analogous to the fact that universal computers sit on all our desks, not an array of specialized devices, one for doing arithmetic, one for word processing, etc.). I suspect there may, indeed, be good reasons to expect that computationally universal mechanisms are out there in biology, and certainly the Cook-Wolfram mathematical result discussed above is a significant piece of any argument toward such a conclusion.

Reviewed by Mark Changizi, Department of Psychological and Brain Sciences, Duke University, Durham, NC 27705; e-mail: changizi@changizi.com.

REFERENCES

1. Penrose, R. Shadows of the Mind; Oxford University Press: New York, 1994.
2. Changizi, M.A. Vagueness, rationality and undecidability: a theory of why there is vagueness. Synthese 1999, 120, 345–374.