A MATHEMATICIAN LOOKS AT WOLFRAM'S NEW KIND OF SCIENCE

LAWRENCE GRAY

1. Overview

In May 2002, Stephen Wolfram finally unveiled his self-proclaimed masterpiece, *A New Kind of Science* (hereinafter referred to as ANKS). Published by Wolfram's own company, the 1280 page volume contains his thoughts on everything from the physics of the universe to the mysteries of human behavior, all based on the results of several years of analyzing the graphical output of some very simple computer programs.

The scope of the book is impressive, covering a bewildering variety of mathematical models, and illustrated by 973 high-resolution black and white pictures. There are whole chapters devoted to biology, physics, and human perception, with shorter sections touching on such unexpected subjects as free will and extraterrestrial art. The extensive historical and technical notes at the end of the book (349 pages of small print) provide fascinating background material.

The primary mathematical focus of the book is a class of discrete-time dynamical systems called *cellular automata*, or "CA's". (See the next section for definitions and examples.) Back in the 1980's, Wolfram introduced several ideas that had a significant impact on CA research, and he also discovered a number of specific CA's with intriguing properties. His activities in this direction were interrupted when he became occupied with the development and promotion of *Mathematica*. But he felt that the ideas in several of his CA papers had never really been "absorbed" by other scientists (ANKS p. 882), so in 1991, he began work on the book that he hopes will start a scientific revolution.

Do we need this revolution? According to Wolfram, "traditional" mathematics and science are doomed — math because of its emphasis on rigorous proof, and science because of its preference for models that can make accurate predictions. He says that the most interesting problems presented by Nature are likely to be formally undecidable or computationally irreducible (ANKS p. 7, pp. 794-795, and p. 1138), rendering proofs and predictions impossible. Mathematicians and scientists have managed to keep busy only by carefully choosing to work on the relatively small set of problems that have simple solutions (ANKS p. 3).

There is more: most mathematical models in science are based on the assumption that time and space are continuous, whereas Wolfram says that time and space are discrete. He would have us abandon models based on calculus and Euclidean geometry in favor of discrete systems like CA's (ANKS p. 8). Indeed, he sees the entire universe as a CA-like system that follows a simple dynamical rule, and the better part of Chapter 9 consists of some clever speculation on the exact nature of such a rule.

To make his argument convincing, Wolfram needed a simple CA that was capable of highly complex behavior. Enter the CA known as "Rule 110", whose dynamical rule is about as simple as possible, as you will see later in this review. The model was discovered by Wolfram in the 1980's, when he conjectured that its behavior was "universal", meaning that it could be used to simulate a universal Turing machine. The conjecture was proved in the 1990's by Matthew Cook, a former employee of Wolfram Research.

Rule 110 is featured prominently throughout ANKS, and it provides the primary motivation for Wolfram's scientific philosophy, which is that the key to understanding complex behavior can be found in very simple discrete systems. He has been pushing this idea for 20 years. But in ANKS, we find a much more provocative version, the "Principle of Computational Equivalence", which we are told is a "new law of nature" that "has vastly richer implications than ... any [other] single collection of laws in science" (ANKS p. 720 and p. 726). The entire final chapter of the book is devoted to this principle, but surprisingly, Wolfram does not provide us with a definitive statement of it. Here is my attempt, pieced together from various phrases in Chapter 12 of ANKS:

> Except for those trajectories that are obviously simple, almost all of the trajectories of a (natural or theoretical) dynamical system can be viewed as computations of equivalent sophistication, which is to say that they are universal. (see ANKS pp. 716-719).

Thus, if you observe a trajectory of some system, such as a CA, or a differential equation, or the weather, or even a bucket of rusting nails, then either you will see something that is obviously simple, or else arbitrarily complex computations will pass before your eyes.

Wolfram's attitude toward traditional mathematics and science is consistent with his Principle. After all, if everything nontrivial behaves like a universal Turing machine, then it is a waste of time to try to find ways to predict anything that is not already obvious. He advises scientists to start doing what he has been doing for the past two decades, which is to systematically explore simple CA's and related discrete systems, searching for models to match various interesting natural phenomena. The Principle of Computational Equivalence seems to suggest that such models are out there somewhere, and Wolfram provides many examples from all areas of science to try to show us that they can actually be found.

Am I convinced? Not really. Wolfram's brand of computer experimentation is a potentially powerful scientific tool. Indeed, I find that by far the most valuable aspect of the book is that it brings together so many interesting examples of CA's and related models that first found the light of day in one of his computer searches. But can he really justify statements like this: "...the new kind of science that I develop in this book is for the first time able to make meaningful statements about even immensely complex behavior" (ANKS p. 3)?

Wolfram loves to tell us why other scientific theories cannot handle complexity. But in these discussions, he badly mischaracterizes his competition. Here is a typical example: "The field of nonlinear dynamics is concerned with analyzing more complicated equations [than linear ones]. Its greatest success has been with so-called soliton equations for which careful manipulation leads to a property similar to linearity. But the kinds of systems that I discuss in this book typically show much more complex behavior, and have no such simplifying properties." (ANKS pp. 15-16). He is particularly hard on chaos theory, which he more or less reduces to a trivial observation about sensitive dependence on initial conditions: "Indeed, all that it shows is that if there is complexity in the details of the initial conditions, then this complexity will eventually appear in the large-scale behavior of the system." He claims to have examples of dynamical systems that exhibit chaotic behavior without sensitive dependence on initial conditions. But his examples are highly questionable, as I will later explain when I discuss his notion of "intrinsic randomness generation".

Wolfram provides very little hard evidence for the Principle of Computational Equivalence. The key phrase "obviously simple" is pretty much left undefined, except to say that it covers systems that are attracted to periodic orbits or follow other obvious patterns. Even when the Principle is taken at face value, serious doubts about both its validity and practical significance have been raised (see the list of reviews given below). I will raise a few more later on, when I discuss fault-tolerant computation and universal CA's.

Despite the provocative attitude and high-minded speculation, there is plenty to enjoy in the book, especially the very accessible and very extensive coverage of so many different kinds of discrete models. In addition to CA's, we find mobile cellular automata, Turing machines, substitution systems, sequential substitution systems, tag systems, cyclic tag systems, register machines, and causal networks. For each type of system, Wolfram presents and carefully explains numerous examples, expertly illustrated by instructive and thought-provoking graphics. I am familiar with Turing machines, for example, but had not seen their workings graphically depicted as in ANKS. Interesting statistics are given about the range of behaviors in these systems, based on Wolfram's own computer experiments.

It is also a lot of fun watching Wolfram find connections with the real world. Some of them are original to Wolfram, many are not, and it is unfortunately not always so easy to determine which is which. But it is great having them all together in one book. Particularly impressive is the chapter where he attempts to capture modern physics, including relativity theory and quantum mechanics, in a CA-like system. Scott Aaronson's review (see below) points out a serious mathematical flaw in this section, and Wolfram's model is based in part on earlier work that is not acknowledged, but the result is impressive and intriguing nonetheless.

In general, the book is easy for the non-expert to read, but difficult for the expert to use. The main part of the text provides a nice non-technical introduction to many topics, but the accompanying notes in the back of the book are hampered by Wolfram's preference for expressing formulas and equations in the language of *Mathematica*. Bibliographic references are entirely lacking, except for a list of Wolfram's own publications. The historical notes are quite thorough, but at the same time, they are heavily biased towards Wolfram's own accomplishments.

My review, like others that have appeared, cannot cover all aspects of ANKS. So I recommend that you also look at the following (given in the approximate order in which I learned of them). All of them can be easily found online.

(1) *Reflections on Stephen Wolfram's "A New Kind of Science"* by Ray Kurzweil. (Ably criticizes the conclusions that Wolfram draws from his Principle of Computational Equivalence.)
(2) Book review by Leo Kadanoff, for *Physics Today*, July 2002. (Provides a balanced perspective on Wolfram's contributions to science, while questioning whether they add up to a whole new kind of science.)

(3) Book review by Scott Aaronson, to appear in *Quantum Information and Computing*, September 2002. (Proves that Wolfram's proposed discrete model for the universe cannot accommodate both special relativity and Bell's Inequality violations.)
(4) Book review by Henry Cohn, for *MAA Online*, June 2002. (Addresses the existence of levels of complexity that lie between the two extremes found in the Principle of Computation Equivalence.)
(5) Book review by Ben Goertzel, for *Extropy*, June 2002. (Amplifies and strengthens Kurzweil's criticisms, and questions Wolfram's rejection of natural selection as a significant factor in evolution.)
(6) *The World According to Wolfram* by Brian Hayes, in *The American Scientist*, July-August 2002. (Counters some of Wolfram's claims of discovery.)
(7) *Blinded by Science* by Jordan Ellenberg, in *Slate*, posted on July 2, 2002 (This is a most entertaining and intelligent review.)
(8) Is the Universe a Universal Computer? by Melanie Mitchell, in *Science*, 4 October 2002. (Takes Wolfram to task for several of his grandiose assertions.)

Each of these articles finds something significant to praise in ANKS (clarity, enthusiasm, expert coverage, fresh perspectives etc.), while at the same time drawing attention to serious difficulties.

In the remainder of my review, I will focus on various mathematical issues raised by Wolfram's presentation of the theory of CA's. I have some familiarity with CA's, since my own area of research centers on their stochastic cousins, known as *probabilistic cellular automata*, or PCA's. My current research concerns PCA models of traffic jams.

2. Cellular Automata

A CA is a deterministic dynamical system, consisting of an array Λ of identical finite machines or *cells* that repeatedly change states or *colors* by following an update rule U. This rule is applied simultaneously to all of the cells in Λ, at discrete time units. When U is applied to a particular cell $x \in \Lambda$, the new color for x is determined by the current colors of the cells in the *neighborhood* of x, denoted by N_x.

Although there are many interesting choices for Λ, I will restrict my attention to the d-dimensional integer lattice $\mathbb{Z}^d$, usually with $d = 1$ or $d = 2$. When $d = 1$, the neighborhood of a cell x is the interval $N_x = \{y \in \mathbb{Z} : |x - y| \leq r\}$, where r is a positive integer parameter called the *range*. When $d = 2$, a common choice is the *Moore neighborhood* with range r, which is the $(2r + 1) \times (2r + 1)$ square block centered at the cell $x = (x_1, x_2)$. More precisely, it is the set

$$N_x = \{y = (y_1, y_2) : |x_1 - y_1| \leq r \text{ and } |x_2 - y_2| \leq r\}.$$

Another choice in the 2-dimensional case is the diamond-shaped *von Neumann neighborhood*:

$$N_x = \{y = (y_1, y_2) : |x_1 - y_1| + |x_2 - y_2| \leq r\}.$$

In general, the cells in a CA can take on one of k different colors, where $k \geq 2$. In my examples, I will take $k = 2$, and I will refer to the 2 colors as 'White' and 'Black'. In pictures, and most verbal descriptions, White will be depicted as □ and Black as ■. But in cases where I want to write down a formula for the update rule U, I will use 0 for White and 1 for Black.

Let us look at two important families of CA's:

The simplest case: $d = 1$**,** $r = 1$**,** $k = 2$**.** This family of CA's is essentially the one that put Wolfram on the map, so it is a good place for us to start. For these CA's, the update rule at a cell x depends on the current colors of the three cells in the neighborhood of x, so that we may denote the new color of x by $U(p, q, r)$, where p, q, r denote the current colors of $x - 1, x, x + 1$ respectively. Since we have $2^3 = 8$ different possible inputs for U and 2 possible outputs, we get $2^8 = 256$ different possible update rules U.

To get an idea of what can happen, we will look at five rules, labeled by Wolfram as Rules 30, 110, 170, 184, and 254. The following table gives the formula for $U(p, q, r)$ in each case. In these formulas, each of the inputs p, q, r can be either a 0 or a 1. The formulas are found by first expressing U as a logical function (using And, Or, Not), and then converting the logical function to a polynomial, using identities like p Or $q = p + q - pq$.

Rule number	$U(p, q, r)$
30	$p + (1 - 2p)(q + r - qr)$
110	$q + r - qr - pqr$
170	r
184	$qr + (1 - q)p$
254	$1 - (1 - p)(1 - q)(1 - r)$

For two of these rules, the behavior is fairly easy to describe. Rule 170 simply shifts the entire sequence of colors to the left, one unit at each update. In Rule 254, ■ never changes to □, and furthermore, ■ spreads from one cell to the next in both directions at each update. So for all initial states except the one with all □'s, Rule 254 goes to the state with all ■'s as time goes to infinity.

Rule 184 is sometimes used as a very simple traffic model, in which ■ represents a car and □ represents a space. At each update, cars move one unit to the right whenever possible. More precisely, if there is a car at cell x and a space at cell $x+1$, then the car moves from x to $x + 1$ at the next update, leaving behind a space at x. This model has been extensively studied, and its behavior is well-understood.

Rule 30 is one of the most interesting CA's ever studied. The first 100 updates for Rule 30 are shown in Figure 2, using the initial state ...□□□■□□□.... You can see this initial state in the top row of the figure, with later states appearing in successive rows, working from the top downward, so that the bottom row in the figure depicts the state after 100 updates. This convention for showing trajectories in 1-dimensional CA's is quite common, and it is the one followed by Wolfram in ANKS.

Towards the left edge of the picture for Rule 30, quite a bit of regularity can be seen. But the rest of the pattern seems quite unpredictable, and there is no known formula for predicting the colors at any given position. In fact, the sequence of colors attained by the cell at the origin forms the basis for the random number generator used by *Mathematica*. This apparently perfectly random behavior was first noticed by Wolfram in 1984, and it seems to defy any sort of rigorous analysis. The best mathematical result that I know about Rule 30 is due to Erica Jen [1], who proved that with the initial state in Figure 2, the sequence of colors attained in any two adjacent cells is not periodic.

Rule 110 is the most interesting of all. With the same initial state as before, the first 750 updates are shown in Figure 2. Note the interesting mixture of chaotic and

FIGURE 1. The first 100 updates for Rule 30

orderly behavior. This CA continues to go through a complicated sequence of states for 2780 updates, after which it finally falls into a predictable pattern. But it turns out that by choosing other initial states, one can get Rule 110 to exhibit arbitrarily complex behavior, in the sense that it can simulate a universal Turing machine. In other words, the Rule 110 CA can be used as a computer that is capable of running any program that can be written in, say, the C++ programming language. To be sure, such a computer would require a rather messy compiler, to convert C++ programs into the corresponding initial states for Rule 110. Run-times would be unbearably long, because the simulation requires all program data to be represented in unary form (rather than binary), forcing an "exponential slowdown". Decoding the program output from the CA trajectories (the "user interface") would also be no piece of cake. But the bottom line is that certain questions about the behavior of Rule 110 are just as hard as the "halting problem" for Turing machines, which is to say that they are undecidable. The only way to try to answer such questions would be to actually compute the sequence of states visited by the CA, one by one. In some cases, the answer would never come.

The universal behavior of Rule 110 was first conjectured by Wolfram around 1985. The result was proved by Matthew Cook in the mid to late 1990's. However, Cook was prevented from publishing this result because he had signed an agreement, as an employee of Wolfram Research, to not disclose his proof until ANKS was published. Because the appearance of ANKS was long-delayed, Cook tested this agreement by presenting his result at a conference at the Santa Fe Institute, at which I was present. His talk was the highlight of the conference, but Wolfram threatened legal action, preventing it from appearing in the conference proceedings. Now that the non-disclosure agreement has expired, Cook's proof should become available. I have seen it and I highly recommend it, because it shows just how tenuous the original conjecture was—the proof just barely works. Of course, Wolfram firmly believes (because of his Principle) that unless a CA is obviously simple (like Rules 170, 184, and 254), it can simulate a universal Turing machine. For example, he believes that Rule 30 is also universal. I find this to be sheer wishful thinking.

The discovery of Rules 30 and 110 is one of Wolfram's greatest contributions to CA theory. Both CA's have very simple rules and very complicated behavior. A significant portion of ANKS is devoted to Rules 30 and 110, and Wolfram speculates

FIGURE 2. The first 750 updates for Rule 110

often that their behavior forms the basis for understanding all that is complex in the world around us.

Outer-totalistic rules. Once we start considering larger neighborhoods, higher dimensions, or more colors, the number of CA rules increases superexponentially. For example, if $d = 2$, $k = 2$, and $r = 1$, with the Moore neighborhood, there are 2^{512} different update rules. It will not be possible to investigate all of these CA's in the lifetime of the universe. So attention has focused on rules that satisfy one or more simplifying assumptions.

Outer-totalistic update rules are those that depend only on the current color of a cell and the sum of the current colors of its neighbors, where the k possible colors are represented by the numbers $\{0, \ldots, k-1\}$. For the case $d = 2$, $k = 2$, and $r = 1$, with the Moore neighborhood, the colors of the 8 neighbors of a cell can sum to anything between 0 and 8, giving 9 possibilities for the sum. There are 2 possibilities for the color of the cell itself, giving 2×9 total possible inputs and 2 possible outputs for the update rule. Thus, there are 2^{18} different outer-totalistic CA's for this case.

In general, for outer-totalistic rules with $k = 2$, it is common to refer to Black cells as *alive* and White cells as *dead.* When a White cell changes to Black, the transition is called a *birth.* The opposite transition is called a *death.* If a live cell does not die during a given update, we say that it *survives.* Outer-totalistic update rules are commonly specified by giving the conditions for the birth and survival of a given cell x. By the definition of outer-totalistic, these conditions depend only on the number of live cells in the neighborhood of x (not counting x itself).

The most famous CA of all is outer-totalistic, with $d = 2$, $k = 2$, and the $r = 1$ Moore neighborhood. It is called the "Game of Life". The birth condition for this rule is that exactly 3 of the 8 neighbors be alive, while the survival condition is that exactly 2 or 3 neighbors be alive.

The Game of Life was discovered in 1970 by John Conway, who was primarily interested in finding a simple CA that could simulate a universal Turing machine. (In Wolfram's version of history, "Conway treated the system largely as a recreation". See ANKS p. 877.) During the next decade, Conway, Gosper, and others discovered that the Game of Life had all of the features that were considered necessary for the Turing machine simulation. A sketch of their argument can be found in the 1982 book *Winning Ways for your Mathematical Plays*, by Berlekamp, Conway, and Guy. Conway's argument was deemed acceptable by Wolfram when he cited it in his paper "Twenty problems in the theory of cellular automata" (1985), but by the time ANKS appeared, his view had changed:

> "The fact remains that a complete and rigorous proof of universality has apparently still never been given for the Game of Life. Particularly in recent years elaborate constructions have been made of for example Turing machines. But so far they have always had only a fixed number of elements on their tape, which is not sufficient for universality." (ANKS p. 1117)

It is hard to know what is meant by this statement. An explicit implementation of a Turing machine in the Game of Life can be found at the website of Paul Rendell. The "tape" in this construction can made arbitrarily long, and can contain up to 8 different symbols on each of its cells. On an infinite lattice, the tape can be infinitely long. Rendell's design for the "head" is expandable to allow for up to 16 states, making it more than adequate for the head of a universal Turing machine. Rendell says his construction was put together "in 1999-2000 mainly using patterns that I created in the 1980's". Rendell's configuration even looks like a Turing machine (the tape and head are clearly visible), and it runs in "real time", up to a constant multiple factor.

In the 1982 book cited above, Conway says: "Life is Universal! ... It's remarkable how such a simple system of genetic rules can lead to such far-reaching results". So it seems that somebody besides Wolfram deserves credit for discovering that simple computer programs can produce highly complex behavior. And yet Wolfram says that this idea is the "pivotal discovery that I made some eighteen years ago", and he considers it to be "one of the more important single discoveries in the whole history of theoretical science" (ANKS p. 2).

There are many other families of CA's in ANKS. There are also a few significant ones that cannot be found there. Outside of ANKS, a good place to start exploring CA's is David Griffeath's website, where you can find a lot of good, genuine mathematics and beautiful pictures.

3. Do the math

Wolfram tells us in ANKS that he has very little use for mathematicians:

> "Over the years, I have watched with disappointment the continuing failure of most scientists and mathematicians to grasp the idea of doing computer experiments on the simplest possible systems ... [Mathematicians] tend to add features to make their systems fit in with complicated and

> abstract ideas — often related to continuity — that exist in modern mathematics. ...One might imagine that the best way to be certain about what could possibly happen in some particular system would be to prove a theorem ...But in my experience ...it is easy to end up making implicit assumptions that can be violated by circumstances one cannot foresee. And indeed, by now, I have come to trust the correctness of conclusions based on simple systematic computer experiments much more than I trust all but the simplest proofs" (ANKS pp. 898-899).

Given this dismissive attitude, it should not be surprising that some of Wolfram's ideas do not hold up so well when examined under the harsh light of rigorous mathematics.

Wolfram's 4 Classes of CA behavior. In 1984, Wolfram introduced a classification scheme for CA's that separated them into 4 Classes. This idea created quite a bit of excitement, and for several years, serious attempts were made to refine the definitions of these classes, and to develop various criteria for determining the class of a CA from numerical or statistical features of its trajectories. Nowadays, these classes remain useful for describing certain general features of CA behavior, but beyond that, they have "proved neither subtle nor fruitful" (to quote Leo Kadanoff). But they play a significant role in ANKS, and some discussion here will be useful.

Rule 254 is a typical Class 1 CA. It has an attracting fixed point, the all ■ state, that attracts all other states, except for one repelling fixed point, which is the all □ state. In general, Class 1 CA's have a single attracting fixed point or periodic orbit, whose basin of attraction is all but a few isolated states. Another way to characterize a Class 1 CA is to say that "information about initial conditions is always rapidly forgotten" (ANKS p. 252).

Rule 170 is in Class 2. Another Class 2 CA is Rule 204, otherwise known as the identity map. A more interesting example is Rule 178, which happens to be outer-totalistic. The birth condition for Rule 178 is that at least one of the two neighbors be alive, and the survival condition is that both neighbors be alive. This rule has two repelling fixed points (all □ and all ■). It has infinitely many states with period 2, in which intervals with □'s at the even cells and ■'s at the odd cells alternate with intervals of the opposite type. From all initial states except the two fixed points, the system rapidly converges to one of the period 2 orbits. In general, a Class 2 CA has many fixed or periodic orbits (possibly modulo a shift, as in Rule 170), and from most initial states, it will quickly converge to one of those orbits. In a Class 2 CA, "some information in the initial state is retained in the final configuration ...but this information always remains completely localized" (ANKS p. 252). CA's in both Class 1 and Class 2 are considered to be "obviously simple" as far as the Principle of Computational Equivalence is concerned.

Rule 30 is a Class 3 CA. This class is characterized by trajectories that have apparent randomness. But not all Class 3 CA's are as unpredictable as Rule 30. Consider, for example, Rule 90, whose update rule is given by $U(p, q, r) = p + r \pmod 2$. Because of the linear nature of this rule, it turns out to be easy to find an explicit formula for any given trajectory of Rule 90. Wolfram places both Rule 30 and Rule 90 into Class 3, primarily on the basis of the visual appearance of their trajectories, which is very similar for most initial states. This appearance reflects the way in which Class 3 CA's "show long-range communication of information —

so that any change made anywhere in the system will almost always eventually be communicated even to the most distant parts of the system" (ANKS p. 252).

Rule 110 is in Class 4, as is the Game of Life. Class 4 is described as being the borderline between Classes 2 and 3, because typical trajectories have regions with apparently random mixing (somewhat like Class 3) and regions with localized structures that either stay stationary or move linearly (somewhat like Class 2). When two such structures collide, various interesting things can happen, as seen in Figure 2, and it is these interactions that make Class 4 special. The proofs of universality for both Rule 110 and the Game of Life make heavy use of the variety that is found in the interactions. Wolfram conjectures (on the basis of his Principle of Computational Equivalence) that all Class 4 CA's are capable of simulating universal Turing machines.

Do these classes exhaust all the possibilities? Wolfram thinks they do. But there are some problems with this view. First of all, there are many CA's that can be assigned to more than one class, depending on the initial state. For example, Rule 184 can act like it is in Class 1 with some initial states, and like it is in Class 2 with others. There are even initial states that make it behave more like a Class 3 CA. Wolfram is aware of this fact (as shown by his discussion of Rule 184 on p. 272 and p. 338), but he mostly ignores it.

A second problem is that some CA's cannot be assigned to any class at all. A good example is the 2-dimensional outer-totalistic CA shown in Figure 3. Births occur if 4, 6, 7, or 8 of the neighbors are alive, and deaths occur if 4, 6, 7, or 8 of the neighbors are dead. Thus, there is a symmetry between the 2 colors. For this example, the initial state is a random uniform mixture of the 2 colors □ and ■, throughout the entire lattice (this is shown in the first picture in Figure 3). Since it is not practical to work on an infinite lattice with such an initial state, the size of the lattice in this example has been restricted to 200×200, with "wraparound" boundary conditions.

The remaining pictures in the figure show the states after 10, 100, and 1000 updates respectively. These pictures and more extensive simulations give convincing evidence for the conjecture that *clustering* occurs. In other words, starting from a very noisy initial state, the system organizes itself into larger and larger regions, each of which primarily contains a single color. This type of behavior does not fit any description I have seen of Classes 1-4. Wolfram discusses this example in ANKS (p. 336), but he does not say anything about how to classify it.

In some ways, the system in Figure 3 is like a Class 3 CA running backwards, because the "long-range communication of information" seems to happen in reverse, as a complicated initial state becomes increasingly simplified through clustering. Perhaps Wolfram would place this CA on the borderline between Classes 1 and 2, because clustering could be interpreted as being midway between being attracted to a single fixed point and being attracted to a large set of fixed or periodic points. But in that case, a new "Class 5" is warranted, analogous to the other borderline case, which is Class 4.

In general, there are lots of interesting types of CA behavior, particularly in 2 dimensions and higher, that are not reflected in the 4 Classes. Yet Wolfram would have us believe that "... at an overall level the behavior we see is not fundamentally much different in two or more dimensions than in one dimension" (ANKS p. 170). I do not find this kind of oversimplification to be very useful.

FIGURE 3. Clustering

In a sense, Wolfram's Principle of Computational Equivalence is an attempt to simplify things even further, by separating all dynamical systems into only 2 classes: "Class A", containing the obviously simple systems, and "Class B", containing the universal ones. All of Class 4 and part of Class 3 (such as Rule 30) are supposed to be in Class B, and everything else is consigned to Class A. Wolfram says there is nothing more complex than Class B. But this view is only partially true, as I will explain in the next two subsections.

Universal CA's. It turns out that there is more than one kind of universality. So far, we have only talked about the kind that primarily concerns Wolfram, which is the ability to simulate arbitrary Turing machines. But there is something known as a *Universal CA*, or UCA, that can do more. A UCA with lattice Λ must be able to simulate every other CA with lattice Λ. Furthermore, the space and time "costs" of the simulation must be bounded above by constant multiples of the space and time requirements of the CA being simulated.

What is the difference between simulating an arbitrary Turing maching and simulating an arbitrary CA? A Turing machine is essentially a computer with a single processor (the head), whereas a CA is a computer with infinitely many parallel processors (the cells). So even a universal Turing machine with an infinite tape cannot actually simulate a CA; it can only simulate larger and larger portions of it, at a slower and slower pace. When a UCA simulates another CA, the cells of the UCA are thought of as being organized into a regular array of "blocks", with each block having the task of simulating a single cell. In this fashion, a UCA can simulate the infinitely many processors (cells) of another CA, using "real time" and "real space" (up to a multiplicative constant), even if the CA being simulated has more colors or a larger range than the UCA. In practice, the distinction is important when one is trying to model something that involves a lot of parallel processing.

Wolfram gives a very nice example of a UCA on pp. 644-656 of ANKS. It is a 1-dimensional CA with $k = 19$ and $r = 1$. We are told that it is possible to reduce the number of colors to 7. The explanation is very clear and well illustrated. But on p. 676 he gets sloppy when he makes the transition from the discussion of UCA's to Rule 110. He never makes the distinction between the two types of universality, making it sound like Rule 110 is a UCA.

Because of certain details about the way in which data is processed in Matthew Cook's Rule 110 Turing simulation, I think it highly unlikely that Rule 110 is a UCA, although I have no proof. On the other hand, there are some explicit constructions, such as David Bell's so-called "Unit Life Cell", that seem to indicate that the Game of Life is a UCA. So there is a precise, mathematical sense in which the Game of Life is capable of performing more sophisticated computations than appear to be possible for Rule 110, contrary to the Principle of Computational Equivalence. Perhaps UCA's should be separated from the other Wolfram classes to form a "Class 6".

The issue of efficient simulation of massively parallel systems is an important one. One of Wolfram's main themes is that complicated mechanisms (such as Darwinian natural selection) are not required for explaining the complexity observed in Nature. To put it simply, he says that mechanisms similar to Rule 110 abound in our physical environment (for example, in chemical reactions), and Rule 110 equals a universal Turing machine, which equals Einstein, so how hard can it be for Nature to produce Einstein? Raymond Kurzweil has done a fine job of attacking this argument, by asking how Nature is able to come up with the "software" needed to make Rule 110 act like Einstein. Ben Goertzel continued the attack by pointing out that Nature could not find the time and space resources to run such software. Rule 110 pretty much has to be abandoned as Wolfram's prime example of a simple system that can explain Nature's complexity.

But what about UCA's like the Game of Life? Presumably they can simulate something highly parallel like the human brain, since the brain could be considered to be some sort of CA on a rather irregular lattice. In the next subsection, I will show that even a well-programmed UCA will not do Nature much good if it wants to produce Einstein.

Fault-tolerant CA's. Even in the most carefully controlled environment, large-scale computing requires sophisticated error-correction. How much more so in the hustle and bustle of our natural world? Most of the known UCA's, such as the Game of Life, are extremely sensitive to errors. Randomly changing the color of a single cell is often enough to turn an elaborate construction like Paul Rendell's Life Turing machine into garbage. It seems highly unlikely that the Game of Life can carry on any sort of non-trivial simulation if it is subjected to random errors, even if the errors are quite rare. In other words, I do not believe that the software exists that can make some natural version of the Game of Life reliably simulate a complex living organism.

A *fault-tolerant universal CA* is a UCA whose ability to simulate other CA's is not affected by random color changes throughout the lattice, provided such "errors" are sufficiently sparse. The existence of such systems was proved by Peter Gacs [2]. His 1-dimensional example is incredibly complicated, and his proof requires more than 200 pages. Gacs has also constructed examples in higher dimensions, where matters are somewhat easier. But his 2-dimensional rule is still far more complicated than, say, the Game of Life. He has a 3-dimensional example (based on ideas of John Reif and Andrei Toom) which is relatively simple. It consists of a 2-dimensional array of synchronized 1-dimensional UCA's that perform mutual error correction. But such a construction does not seem like something that would easily arise in Nature.

Thus, it is not clear that Nature could easily find a mechanism that behaves like a fault-tolerant UCA, without some "guiding hand" like natural selection. But Wolfram's thesis seems to require Nature to do just that. Until someone finds a simple fault-tolerant UCA, this particular idea in ANKS remains wishful speculation.

At any rate, I consider fault-tolerant UCA's to be more powerful and sophisticated than ordinary UCA's, which are likely to be more powerful than Rule 110. I propose that we assign fault-tolerant UCA's to "Class 7". There is even a further level of computational sophistication, Gacs's so-called "self-organizing" fault-tolerant UCA's, which may also be important to Nature. Life is not necessarily so simple after all!

Metastability. "Finite size effects" are the bane of computer experimentation with CA's. It is not unusual to observe a particular behavior in a CA computer simulation, only to find out much later that this behavior evaporates when the simulation is run on a computer with larger resources (space, time, or both). In the interim, a lot of time can be wasted theorizing about the illusory behavior that seemed so intriguing in the smaller system.

The computer naturally restricts the lattice in any simulation to a finite size. For 1-dimensional systems, it is possible to run simulations on very large finite lattices, and conclusions based on such experiments can be reasonably reliable. But for 2-dimensional systems, even simulations with modest lattice sizes like 1000×1000 can become unwieldy, and many phenomena require considerably larger lattices for reliable observation. Furthermore, the effects of "boundary conditions" (how to define the update rule at the edges of the finite lattice) can be quite pronounced in 2 dimensions. Thus, it is not too surprising that most of the examples in Wolfram's book are either 1-dimensional, or 2-dimensional with fairly small lattice sizes. Systems in 3 or higher dimensions receive very little attention.

Here is a simple example of a finite size effect in 2 dimensions that actually fooled some experimentalists. It is an outer-totalistic CA with 2 colors, using the range 1 von Neumann neighborhood. The birth rule is that at least 2 of the 4 neighbors be alive, and the survival rule is that all live cells survive. This model is called *bootstrap percolation*. It is not discussed in ANKS in spite of its importance.

If the initial population of live cells in bootstrap percolation is random but sparse, the system seems to rapidly converge to a fixed point, consisting of various rectangular "islands" of live cells, surrounded by a "sea" of dead cells. Run 100 experiments with this system on a 1000×1000 lattice, using initial states in which the live cells are randomly distributed with density $p = .03$, and you are likely to see this same Class 2 behavior over and over. Usually the system finds a fixed point within about 60 updates.

When the initial density is changed to $p = .05$ on a 1000×1000 lattice, we usually see Class 1 behavior. One or more of the islands continues to grow, feeding off of "debris" that consists of smaller islands and individual live cells. Eventually, the system reaches the fixed point in which every cell is alive. This phenomenon led some experimentalists to believe that there is a critical value of p (in this case, somewhere between .03 and .05) that separates the two types of behavior. They were wrong. The Class 2 behavior observed for small p is only metastable; it is a finite size effect. The threshold for p goes to 0 as the lattice size L goes to infinity. The system was carefully analyzed by Michael Aizenman and Joel Lebowitz [3], who gave good quantitative information about the amount of space and time that is required to see the Class 1 behavior. More recently, Alexander Holroyd [6] proved a much more precise result, which is that the asymptotic threshold occurs when $p \to 0$ and $L \to \infty$ in such a way that $\lim p \log L = \pi^2/18$.

In retrospect, it is obvious that the Class 2 behavior of bootstrap percolation is a finite size effect, because there is a very simple argument, based on the Borel-Cantelli Lemma from probability theory, that in a large enough lattice, a sufficiently large island will appear that will continue to grow without bound. But other finite size effects are not always so easy to detect.

Has Wolfram ever been fooled by a finite size effect? Possibly. Consider the following 2-dimensional outer-totalistic model. It has 2 colors and uses the range 1 Moore neighborhood. The birth condition is the same as for the Game of Life (exactly 3 of the 8 cells must be alive), but the rule for survival is different: a live cell survives only if no more than 4 of its 8 neighbors are also alive. This example is featured on p. 178 of ANKS, although Wolfram's description there of the update rule is unfortunately incomplete. (This is not the only error that I have found in his descriptions of CA rules: one of the formulas that he gives for Rule 110 on page 869 is also incorrect.)

Wolfram says that this CA exhibits a growth pattern whose "shape closely approximates a circle", or to be more precise, a disk. The picture in ANKS depicts the state of the system after 400 updates, starting from an initial state in which the "finite seed" ■■■■■■■ is surrounded by an infinite sea of □'s. I have run this example for several thousand more updates, and the disk-shaped pattern seems to continue to grow indefinitely. Similar behavior is found using many other such seeds.

But matters are not as simple as one might think. If we start the system with the seed ■■■□■, we observe the familiar growing disk shape for roughly 2700 updates. Then something strange happens. At the left boundary of the disk, a small but very distinct peak suddenly appears. This peak soon dominates the left side of the disk, and it appears to be a permanent feature.

What happened? The pattern of live cells around the edge of the growing shape fluctuates quite unpredictably. It happens that somewhere around 2750 updates, these fluctuations produce a rare configuration that generates a "spike" that grows rapidly, sticking straight out from the disk. As the spike lengthens, the area around it is filled in, and the peak is formed. Figure 3 shows a closeup of the peak region. I learned about the seed for this example from David Griffeath, who got it from Matthew Cook. As far as I know, Wolfram did not know about this seed when ANKS was published.

Will a similar peak eventually appear if we use Wolfram's seed? No one knows. I have run the system from that seed for more than 10,000 updates without seeing anything. After that, my computer slows down to a crawl. But I conjecture that the wild fluctuations around the edge of the shape must eventually produce the special pattern needed for the peak-producing spike, provided they do not first create some other strange feature.

Wolfram first considered the possibility of circular growth in his paper with Norman Packard, "Two-dimensional cellular automata" (1984). The examples in that paper were not very convincing, because the shapes were so rough. The example in ANKS is supposed to be an improvement. Wolfram's experiments convinced him that he had finally found a CA that exhibited an asymptotic growth pattern that

FIGURE 4. An unexpected peak at the left edge of a growing disk

was nearly perfectly circular. But it now seems likely that his conclusion is wrong. Computer experiments cannot be trusted.

Intrinsic randomness generation. Wolfram has a lot to say about randomness in ANKS. Much of Chapter 10 is devoted to the discussion of his practical definition of randomness, which is more or less that something is random if it looks random and passes the standard statistical tests for randomness. Anyone who has used a pseudo-random number generator in computer simulations should have some sympathy for this approach, even though it ignores interesting recent work that brings mathematical rigor to the notion of pseudo-randomness (for an introduction, see [7]).

Wolfram also gives a lot of attention to something he calls *intrinsic randomness generation*, which I will refer to as "IRG". This idea first appeared in his 1985 paper entitled "Origins of randomness in physical systems", and it is one of those that he says were not adequately understood by the rest of the scientific community.

According to Wolfram, IRG is one of three different "sources" of randomness in dynamical systems. The other two are (i) randomness in the initial conditions, which he says is the primary preoccupation of chaos theory, and (ii) random noise in the environment, which can be considered the reason for stochastic process theory. These two types of randomness are considered to come from "outside" of a dynamical system. IRG is something that is supposed to arise "within" a deterministic dynamical system, without any help from the outside. Wolfram speculates that since the universe is a deterministic CA-like system, all randomness must be ultimately traceable to IRG. And, of course, he claims IRG as his own discovery.

Rule 30 provides Wolfram's favorite example of IRG. It is a deterministic system, so random noise is not an issue. And, as seen in Figure 2, it does not seem to need randomness in the initial conditions to behave in an apparently random fashion. By all of the standard tests, Rule 30 produces better random sequences than any of the other pseudo-random number generators in common use, and Wolfram has every right to be proud of it.

But in what way is IRG different from, say, the kind of behavior that can be observed in models familiar from chaos theory? Here is Wolfram's answer:

> "...How can one tell in an actual experiment on some system in nature to what extent intrinsic randomness generation is really the mechanism

> responsible? ... The clearest sign is a somewhat unexpected phenomenon: ... if intrinsic randomness generation is ... at work, then the precise details of the behavior can ... be repeatable" (ANKS p. 323)

The subsequent discussion makes it clear that Wolfram intends this repeatability to be present even if small random perturbations are made to the system, as will inevitably happen in physical experiments.

The point of this criterion is to make a clear distinction between IRG and the kind of unpredictable behavior typically found in chaos theory, which is always closely linked to "sensitive dependence". In chaos theory, sensitive dependence is not sufficient for chaotic behavior, but it has always been considered necessary. Wolfram's repeatability criterion implies that sensitive dependence is not necessary when IRG is the dominant force.

If Wolfram could produce a legitimate example of this kind of repeatability, then he would have something very surprising. And indeed, he claims to have both theoretical and physical examples. Unfortunately, his one theoretical example contains a fatal mathematical error (as I will explain shortly).

Wolfram mentions a few physics and biology experiments (ANKS p. 976) that are supposed to demonstrate repeatably random behavior, but since no references are given, we are forced to take his word for it. In his 1985 paper, there is an actual reference to a physics experiment that was run 5 times with nominally identical initial conditions. For an initial time segment, there are two different random-looking outputs, one that is shared by three of the runs, and another that is shared by two of the runs. After the initial time period, all 5 outputs diverge from one another. This is a sort of weak repeatability that is interesting, but it is not a completely convincing example of Wolfram's idea. A plausible explanation is given in the experimental report that reminds me somewhat of a phenomenon observed in models of billiards, where trajectories with similar initial conditions can stay close to one another until some crisis (a collision) causes them to diverge. Such models are considered to have sensitive dependence, as Wolfram himself explains with a similar example involving mirrors on p. 311 of ANKS. So I do not buy into Wolfram's physical examples.

For his theoretical example of repeatability, Wolfram naturally turns to Rule 30. He admits that this model does in fact exhibit sensitive dependence (and hence non-repeatability) when it is perturbed by randomly changing the color of one or more cells. But then he introduces a second "less drastic" kind of perturbation, and finds that the trajectories of Rule 30 are repeatable under such perturbations. Unfortunately, it turns out that these perturbations are so mild that repeatability occurs for *every* CA, as I will now show.

For CA's having the 2 colors □ = 0 and ■ = 1, Wolfram's perturbation works as follows. First enlarge the color "palette" to be an interval I, producing a continuous "gray scale". Then extend the update rule U to all of I in some smooth way. For example, one can use a polynomial expression for U, as in Table 1. Each update is now performed in two steps: first, the color at each cell is randomly perturbed by a small amount, and then the extended update rule is applied.

This "less drastic" perturbation scheme would be convincing, except for the fact that Wolfram has a very special way of extending the update rule. His extended rule takes the form $f \circ V$, where V is a polynomial version of the original update rule U, and f is a smooth function on $\mathbb{R}$ with fixed points at 0 and 1. Furthermore,

$f'(0) = f'(1) = 0$. (Actually, I have simplified things slightly from what Wolfram does. But my basic argument is still valid for his version.) So, in a sense, the update is actually performed in three steps: first do the random perturbation, then apply V, then apply f.

It is this extra step involving f that invalidates the procedure as any sort of useful criterion. Using the properties of f, it is easy to prove that for any given V, the size of the perturbations can be chosen small enough so that after f is applied, the result is always extremely close to 0 and 1. In effect, the fixed points of f are so stable that they virtually nullify the perturbations before they have any chance to impact the behavior of the system. The nature of the underlying CA rule is irrelevant.

Thus, ANKS contains no mathematical example of "intrinsically random" behavior without sensitive dependence, and the physical examples are either uncheckable or unconvincing. Rule 30 is merely a very good example of a dynamical system that has chaotic trajectories and sensitive dependence. The fact that some of these trajectories have "simple" initial conditions is interesting, but not at all unprecedented. There are many well-known examples of simple dynamical systems with "chaotic attractors". The basin of attraction for such attractors can include a fairly large set, so that many trajectories with simple initial conditions converge to motion along the chaotic attractor, and such motion is quite unpredictable. If this is IRG, then Wolfram has merely "discovered" a fancy new name for a well-known phenomenon.

In ANKS, Wolfram says that " ...the core of this book can be viewed as introducing a major generalization of mathematics" (p. 7). In this he is entirely mistaken, but there are at least two ways in which he has benefited mathematics: he has helped to popularize a relatively little known mathematical area (CA theory), and he has unwittingly provided several highly instructive examples of the pitfalls of trying to dispense with mathematical rigor.

4. For further study

The most immediate source for learning more about CA's is David Griffeath's website, `psoup.math.wisc.edu`. There you will find numerous links to other web resources, including software for running CA simulations. For recent progress in explicit implementations of universal computation, see Paul Chapman's Game of Life version of a "Universal Minsky Register Machine" at `www.igblan.com/ca`. A classic book on CA's is [5]. It was written by two pioneers in the field, and has a perspective that is quite different from Wolfram's. There is a lot more to CA theory than can be found in ANKS, as you can determine for yourself by randomly browsing the web (for example, try investigating "cyclic cellular automata", which are not mentioned in ANKS). Wolfram's papers on CA's and complexity theory can be found at his website `www.stephenwolfram.com`. Another Wolfram website, `www.wolframscience.com`, contains a growing set of "interactive tools" that supplement ANKS.

I benefited from conversations with many mathematicians, including David Griffeath, Peter Gacs, Rick Moeckel, and Bill Casselman. David and Bill also helped me considerably with the figures, which were generated using the freely available software MCell and Life32. I am also grateful for several comments contributed by readers of the online version of this article.

References

[1] Erica Jen, *Aperiodicity in One-dimensional Cellular Automata*, Physica D **45** (1990), 3–18.

[2] Peter Gacs, *Reliable Cellular Automata with Self-organization*, Journal of Statistical Physics **103** (2001), 45–267.

[3] Michael Aizenman and Joel Lebowitz, *Metastability Effects in Bootstrap Percolation*, Journal of Physics A: Mathematical and General **21** (1988), 3801–3813.

[4] Stephen Wolfram, *Cellular Automata and Complexity: Collected Papers*, Addison-Wesley (1994).

[5] Tommaso Toffoli and Norman Margolis, *Cellular Automata Machines*, The MIT Press (1987).

[6] Alexander Holroyd, *Sharp Metastability Threshold for Two-Dimensional Bootstrap Percolation*, To appear in Probability and Related Fields.

[7] Oded Goldreich, *Fundamentals of Cryptography*, Cambridge University Press (2001).

School of Mathematics, University of Minnesota, Minneapolis, MN 55455

E-mail address: `gray@math.umn.edu`