EXCERPTED FROM

STEPHEN WOLFRAM

A NEW KIND OF SCIENCE

SECTION 3.12

How the Discoveries in This Chapter Were Made

there are general principles that govern the behavior of a wide range of systems, independent of the precise details of each system.

And it is this that means that even if we do not know all the details of what is inside some specific system in nature, we can still potentially make fundamental statements about its overall behavior. Indeed, in most cases, the important features of this behavior will actually turn out to be ones that we have already seen with the various kinds of very simple rules that we have discussed in this chapter.

How the Discoveries in This Chapter Were Made

This chapter—and the last—have described a series of surprising discoveries that I have made about what simple programs typically do. And in making these discoveries I have ended up developing a somewhat new methodology—that I expect will be central to almost any fundamental investigation in the new kind of science that I describe in this book.

Traditional mathematics and the existing theoretical sciences would have suggested using a basic methodology in which one starts from whatever behavior one wants to study, then tries to construct examples that show this behavior. But I am sure that had I used this approach, I would not have got very far. For I would have looked only for types of behavior that I already believed might exist. And in studying cellular automata, this would for example probably have meant that I would only have looked for repetition and nesting.

But what allowed me to discover much more was that I used instead a methodology fundamentally based on doing computer experiments.

In a traditional scientific experiment, one sets up a system in nature and then watches to see how it behaves. And in much the same way, one can set up a program on a computer and then watch how it behaves. And the great advantage of such an experimental approach is that it does not require one to know in advance exactly what kinds of behavior can occur. And this is what makes it possible to discover genuinely new phenomena that one did not expect.

Experience in the traditional experimental sciences might suggest, however, that experiments are somehow always fundamentally imprecise.

For when one deals with systems in nature it is normally impossible to set up or measure them with perfect precision—and indeed it can be a challenge even to make a traditional experiment be at all repeatable.

But for the kinds of computer experiments I do in this book, there is no such issue. For in almost all cases they involve programs whose rules and initial conditions can be specified with perfect precision—so that they work exactly the same whenever and wherever they are run.

In many ways these kinds of computer experiments thus manage to combine the best of both theoretical and experimental approaches to science. For their results have the kind of precision and clarity that one expects of theoretical or mathematical statements. Yet these results can nevertheless be found purely by making observations.

Yet as with all types of experiments it requires considerable skill and judgement to know how to set up a computer experiment that will yield meaningful results. And indeed, over the past twenty years or so my own methodology for doing such experiments has become vastly better.

Over and over again the single most important principle that I have learned is that the best computer experiments are ones that are as simple and straightforward as possible. And this principle applies both to the structure of the actual systems one studies—and to the procedures that one uses for studying them.

At some level the principle of looking at systems with the simplest possible structure can be viewed as an abstract aesthetic one. But it turns out also to have some very concrete consequences.

For a start, the simpler a structure is, the more likely it is that it will show up in a wide diversity of different places. And this means that by studying systems with the simplest possible structure one will tend to get results that have the broadest and most fundamental significance.

In addition, looking at systems with simpler underlying structures gives one a better chance of being able to tell what is really responsible for any phenomenon one sees—for there are fewer features that have been put into the system and that could lead one astray.

At a purely practical level, there is also an advantage to studying systems with simpler structures; for these systems are usually easier to implement on a computer, and can thus typically be investigated more extensively with given computational resources.

But an obvious issue with saying that one should study systems with the simplest possible structure is that such systems might just not be capable of exhibiting the kinds of behavior that one might consider interesting—or that actually occurs in nature.

And in fact, intuition from traditional science and mathematics has always tended to suggest that unless one adds all sorts of complications, most systems will never be able to exhibit any very relevant behavior. But the results so far in this book have shown that such intuition is far from correct, and that in reality even systems with extremely simple rules can give rise to behavior of great complexity.

The consequences of this fact for computer experiments are quite profound. For it implies that there is never an immediate reason to go beyond studying systems with rather simple underlying rules. But to absorb this point is not an easy matter. And indeed, in my experience the single most common mistake in doing computer experiments is to look at systems that are vastly more complicated than is necessary.

Typically the reason this happens is that one just cannot imagine any way in which a simpler system could exhibit interesting behavior. And so one decides to look at a more complicated system—usually with features specifically inserted to produce some specific form of behavior.

Much later one may go back and look at the simpler system again. And this is often a humbling experience, for it is common to find that the system does in fact manage to produce interesting behavior—but just in a way that one was not imaginative enough to guess.

So having seen this many times I now always try to follow the principle that one can never start with too simple a system. For at worst, one will just establish a lower limit on what is needed for interesting behavior to occur. But much more often, one will instead discover behavior that one never thought was possible.

It should however be emphasized that even in an experiment it is never entirely straightforward to discover phenomena one did not expect. For in setting up the experiment, one inevitably has to make assumptions about the kinds of behavior that can occur. And if it turns out that there is behavior which does not happen to fit in with these assumptions, then typically the experiment will fail to notice it.

In my experience, however, the way to have the best chance of discovering new phenomena in a computer experiment is to make the design of the experiment as simple and direct as possible. It is usually much better, for example, to do a mindless search of a large number of possible cases than to do a carefully crafted search of a smaller number. For in narrowing the search one inevitably makes assumptions, and these assumptions may end up missing the cases of greatest interest.

Along similar lines, I have always found it much better to look explicitly at the actual behavior of systems, than to work from some kind of summary. For in making a summary one inevitably has to pick out only certain features, and in doing this one can remove or obscure the most interesting effects.

But one of the problems with very direct experiments is that they often generate huge amounts of raw data. Yet what I have typically found is that if one manages to present this data in the form of pictures then it effectively becomes possible to analyze very quickly just with one's eyes. And indeed, in my experience it is typically much easier to recognize unexpected phenomena in this way than by using any kind of automated procedure for data analysis.

It was in a certain sense lucky that one-dimensional cellular automata were the first examples of simple programs that I investigated. For it so happens that in these systems one can usually get a good idea of overall behavior just by looking at an array of perhaps 10,000 cells—which can easily be displayed in few square inches.

And since several of the 256 elementary cellular automaton rules already generate great complexity, just studying a couple of pages of pictures like the ones at the beginning of this chapter should in principle have allowed one to discover the basic phenomenon of complexity in cellular automata.

But in fact I did not make this discovery in such a straightforward way. I had the idea of looking at pictures of cellular automaton evolution at the very beginning. But the technological difficulty of producing these pictures made me want to reduce their number as much as possible. And so at first I looked only at the 32 rules which had left-right symmetry and made blank backgrounds stay unchanged.

Among these rules I found examples of repetition and nesting. And with random initial conditions, I found more complicated behavior. But since I did not expect that any complicated behavior would be possible with simple initial conditions, I did not try looking at other rules in an attempt to find it. Nevertheless, as it happens, the first paper that I published about cellular automata—in 1983—did in fact include a picture of rule 30 from page 27, as an example of a non-symmetric rule. But the picture showed only 20 steps of evolution, and at the time I did not look carefully at it, and certainly did not appreciate its significance.

For several years, I did progressively more sophisticated computer experiments on cellular automata, and in the process I managed to elucidate many of their properties. But finally, when technology had advanced to the point where it became almost trivial for me to do so, I went back and generated some straightforward pages of pictures of all 256 elementary rules evolving from simple initial conditions. And it was upon seeing these pictures that I finally began to appreciate the remarkable phenomenon that occurs in systems like rule 30.

Seven years later, after I had absorbed some basic intuition from looking at cellular automata like rule 30, I resolved to find out whether similar phenomena also occurred in other kinds of systems. And the first such systems that I investigated were mobile automata.

Mobile automata in a sense evolve very slowly relative to cellular automata, so to make more efficient pictures I came up with a scheme for showing their evolution in compressed form. I then started off by generating pictures of the first hundred, then the first thousand, then the first ten thousand, mobile automata. But in all of these pictures I found nothing beyond repetitive and nested behavior.

Yet being convinced that more complicated behavior must be possible, I decided to persist, and so I wrote a program that would automatically search through large numbers of mobile automata. I set up various criteria for the search, based on how I expected mobile automata could behave. And quite soon, I had made the program search a million mobile automata, then ten million.

But still I found nothing.

So then I went back and started looking by eye at mobile automata with large numbers of randomly chosen rules. And after some time what I realized was that with the compression scheme I was using there could be mobile automata that would be discarded according to my search criteria, but which nevertheless still had interesting behavior. And within an hour of modifying my search program to account for this, I found the example shown on page 74.

Yet even after this, there were still many assumptions implicit in my search program. And it took some time longer to identify and remove them. But having done so, it was then rather straightforward to find the example shown on page 75.

A somewhat similar pattern has been repeated for most of the other systems described in this chapter. The main challenge was always to avoid assumptions and set up experiments that were simple and direct enough that they did not miss important new phenomena.

In many cases it took a large number of iterations to work out the right experiments to do. And had it not been for the ease with which I could set up new experiments using *Mathematica*, it is likely that I would never have gotten very far in investigating most of the systems discussed in this chapter. But in the end, after running programs for a total of several years of computer time—corresponding to more than a million billion logical operations—and creating the equivalent of tens of thousands of pages of pictures, I was finally able to find all of the various examples shown in this chapter and the ones that follow.