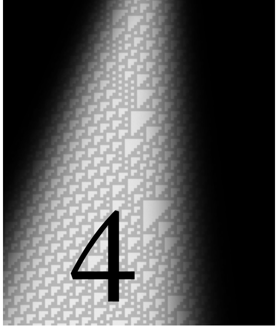
EXCERPTED FROM

STEPHEN WOLFRAM
A NEW KIND OF SCIENCE

SECTION 4.1

The Notion of Numbers



Systems Based on Numbers

The Notion of Numbers

Much of science has in the past ultimately been concerned with trying to find ways to describe natural systems in terms of numbers.

Yet so far in this book I have said almost nothing about numbers. The purpose of this chapter, however, is to investigate a range of systems that are based on numbers, and to see how their behavior compares with what we have found in other kinds of systems.

The main reason that systems based on numbers have been so popular in traditional science is that so much mathematics has been developed for dealing with them. Indeed, there are certain kinds of systems based on numbers whose behavior has been analyzed almost completely using mathematical methods such as calculus.

Inevitably, however, when such complete analysis is possible, the final behavior that is found is fairly simple.

So can systems that are based on numbers ever in fact yield complex behavior? Looking at most textbooks of science and mathematics, one might well conclude that they cannot. But what one must realize is that the systems discussed in these textbooks are usually ones that are specifically chosen to be amenable to fairly complete analysis, and whose behavior is therefore necessarily quite simple.

And indeed, as we shall see in this chapter, if one ignores the need for analysis and instead just looks at the results of computer experiments, then one quickly finds that even rather simple systems based on numbers can lead to highly complex behavior.

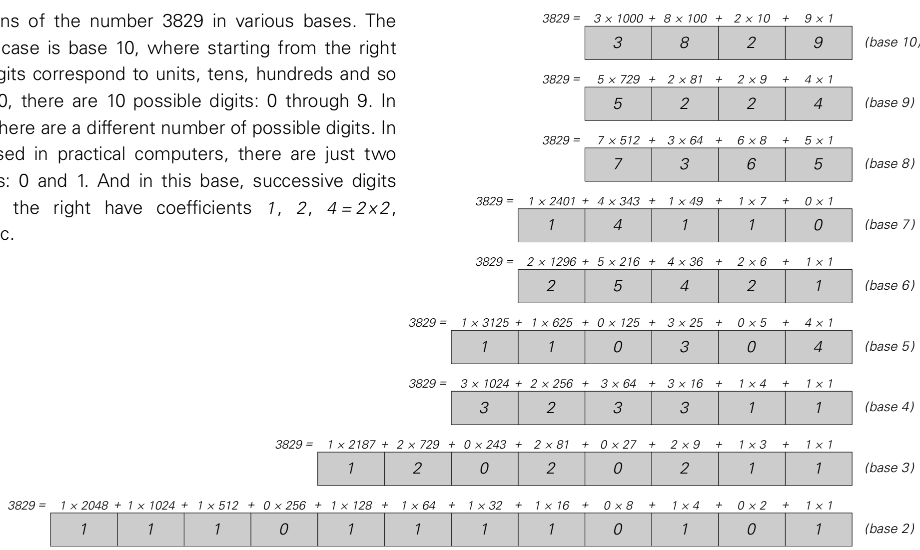
But what is the origin of this complexity? And how does it relate to the complexity we have seen in systems like cellular automata?

One might think that with all the mathematics developed for studying systems based on numbers it would be easy to answer these kinds of questions. But in fact traditional mathematics seems for the most part to lead to more confusion than help.

One basic problem is that numbers are handled very differently in traditional mathematics from the way they are handled in computers and computer programs. For in a sense, traditional mathematics makes a fundamental idealization: it assumes that numbers are elementary objects whose only relevant attribute is their size. But in a computer, numbers are not elementary objects. Instead, they must be represented explicitly, typically by giving a sequence of digits.

The idea of representing a number by a sequence of digits is familiar from everyday life: indeed, our standard way of writing numbers corresponds exactly to giving their digit sequences in base 10. What base 10 means is that for each digit there are 10 possible choices:

Representations of the number 3829 in various bases. The most familiar case is base 10, where starting from the right successive digits correspond to units, tens, hundreds and so on. In base 10, there are 10 possible digits: 0 through 9. In other bases, there are a different number of possible digits. In base 2, as used in practical computers, there are just two possible digits: 0 and 1. And in this base, successive digits starting from the right have coefficients 1, 2, $4 = 2 \times 2$, $8 = 2 \times 2 \times 2$, etc.



0 through 9. But as the picture at the bottom of the facing page shows, one can equally well use other bases. And in practical computers, for example, base 2 is almost always what is used.

So what this means is that in a computer numbers are represented by sequences of 0's and 1's, much like sequences of white and black cells in systems like cellular automata. And operations on numbers then correspond to ways of updating sequences of 0's and 1's.

In traditional mathematics, the details of how operations performed on numbers affect sequences of digits are usually considered quite irrelevant. But what we will find in this chapter is that precisely by looking at such details, we will be able to see more clearly how complexity develops in systems based on numbers.

In many cases, the behavior we find looks remarkably similar to what we saw in the previous chapter. Indeed, in the end, despite some confusing suggestions from traditional mathematics, we will discover that the general behavior of systems based on numbers is very similar to the general behavior of simple programs that we have already discussed.

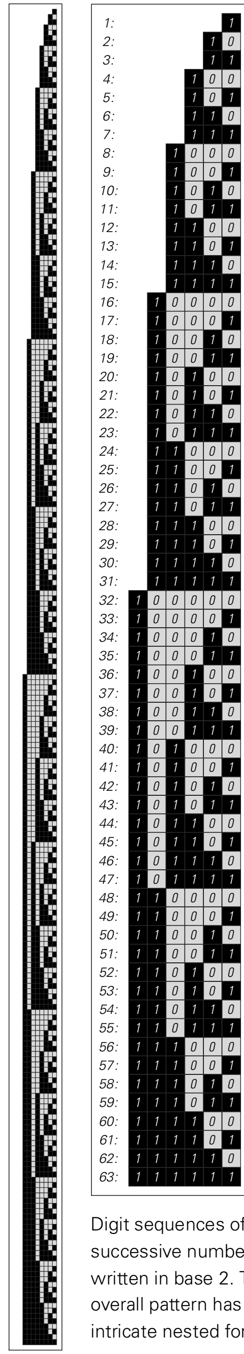
Elementary Arithmetic

The operations of elementary arithmetic are so simple that it seems impossible that they could ever lead to behavior of any great complexity. But what we will find in this section is that in fact they can.

To begin, consider what is perhaps the simplest conceivable arithmetic process: start with the number 1 and then just progressively add 1 at each of a sequence of steps.

The result of this process is to generate the successive numbers 1, 2, 3, 4, 5, 6, 7, 8, ... The sizes of these numbers obviously form a very simple progression.

But if one looks not at these overall sizes, but rather at digit sequences, then what one sees is considerably more complicated. And in fact, as the picture on the right demonstrates, these successive digit sequences form a pattern that shows an intricate nested structure.



Digit sequences of successive numbers written in base 2. The overall pattern has an intricate nested form.