EXCERPTED FROM

STEPHEN WOLFRAM
A NEW KIND OF SCIENCE

SECTION 11.2

Computations in Cellular Automata

abstract terms about the computation that is performed, without necessarily looking at all the details of how it actually works.

Why is such an abstraction useful? The main reason is that it potentially allows one to discuss in a unified way systems that have completely different underlying rules. For even though the internal workings of two systems may have very little in common, the computations the systems perform may nevertheless be very similar.

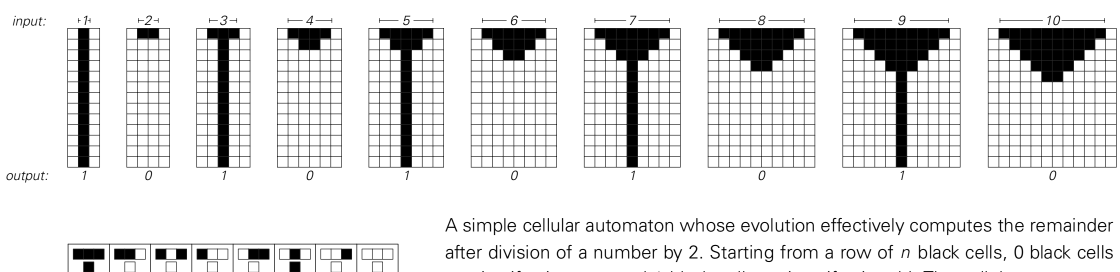
And by thinking in terms of such computations, it then becomes possible to imagine formulating principles that apply to a very wide variety of different systems—quite independent of the detailed structure of their underlying rules.

Computations in Cellular Automata

I have said that the evolution of a system like a cellular automaton can be viewed as a computation. But what kind of computation is it, and how does it compare to computations that we typically do in practice?

The pictures below show an example of a cellular automaton whose evolution can be viewed as performing a particular simple computation.

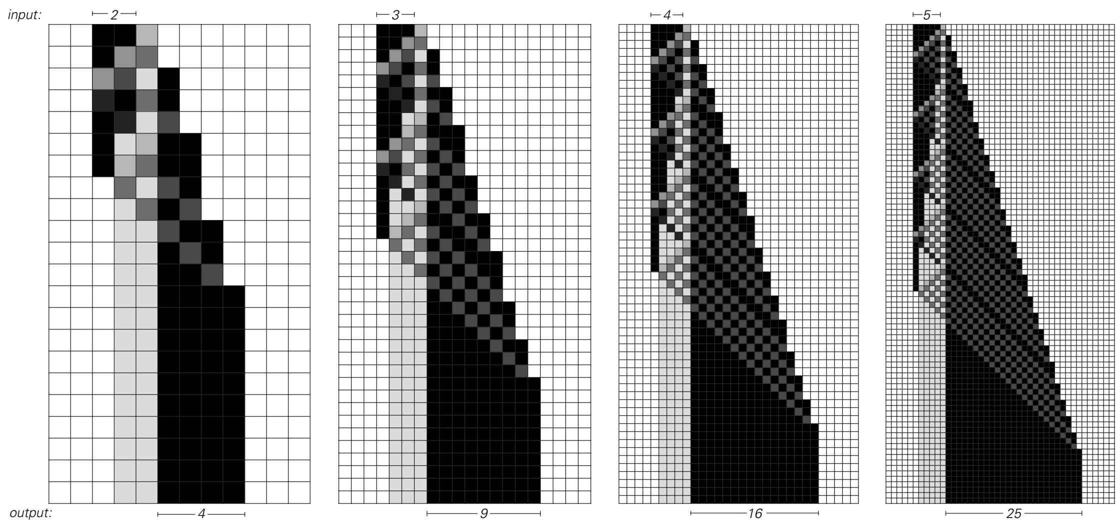
If one starts this cellular automaton with an even number of black cells, then after a few steps of evolution, no black cells are left. But if instead one starts it with an odd number of black cells, then a single black cell survives forever. So in effect this cellular automaton can be viewed as computing whether a given number is even or odd.



A simple cellular automaton whose evolution effectively computes the remainder after division of a number by 2. Starting from a row of *n* black cells, 0 black cells survive if *n* is even, and 1 black cell survives if *n* is odd. The cellular automaton follows elementary rule 132, as shown on the left.

One specifies the input to the computation by setting up an appropriate number of initial black cells. And then one determines the result of the computation by looking at how many black cells survive in the end.

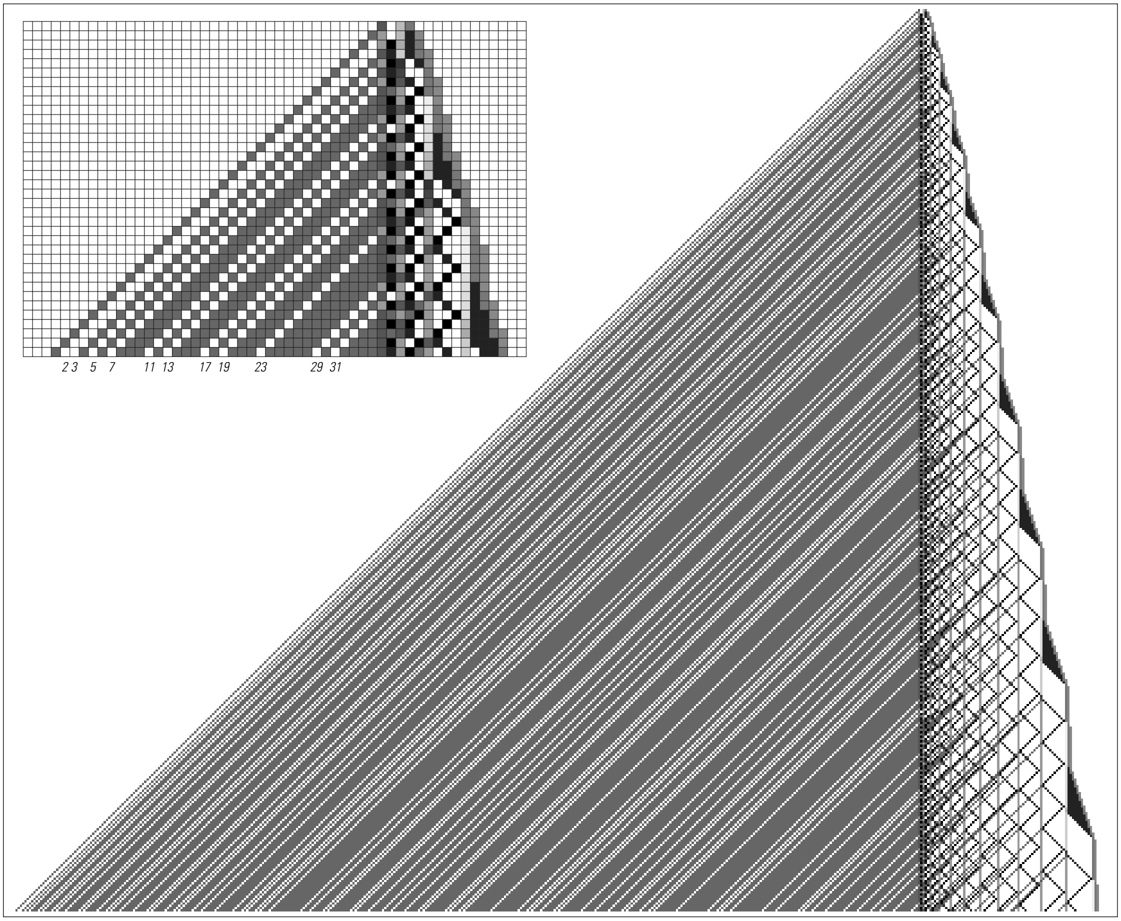
Testing whether a number is even or odd is by most measures a rather simple computation. But one can also get cellular automata to do more complicated computations. And as an example the pictures below show a cellular automaton that computes the square of any number. If one starts say with 5 black squares, then after a certain number of steps the cellular automaton will produce a block of exactly $5 \times 5 = 25$ black squares.



A cellular automaton that computes the square of any number. The cellular automaton effectively works by adding the original number n together n times. The underlying rule used here involves eight possible colors for each cell.

At first it might seem surprising that a system with the simple underlying structure of a cellular automaton could ever be made to perform such a computation. But as we shall see later in this chapter, cellular automata can in fact perform what are in effect arbitrarily sophisticated computations. And as one example of a somewhat more sophisticated computation, the picture on the next page shows a cellular automaton that computes the successive prime numbers: 2, 3, 5, 7, 11, 13, 17, etc.

The rule for this cellular automaton is somewhat complicated—it involves a total of sixteen colors possible for each cell—but the example demonstrates the point that in principle a cellular automaton can compute the primes.



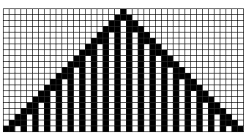
A cellular automaton constructed to compute the prime numbers. The system generates a dark gray stripe on the left at all positions that correspond to any product of numbers other than 1. White gaps then remain at positions that correspond to the prime numbers 2, 3, 5, 7, 11, 13, 17, etc. The cellular automaton effectively does its computation using the standard sieve of Eratosthenes method. The structures on the right bounce backwards and forwards with repetition periods corresponding to successive odd numbers. Once in each period they produce a gray stripe which propagates to the left, so that in the end there is a gray stripe corresponding to every multiple of every number. The rule for the cellular automaton shown here involves 16 possible colors for each cell.

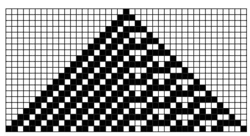
So what about the cellular automata that we discussed earlier in this book? What kinds of computations can they perform?

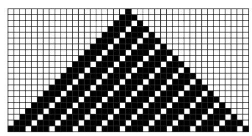
At some level, any cellular automaton—or for that matter, any system whatsoever—can be viewed as performing a computation that determines what its future behavior will be.

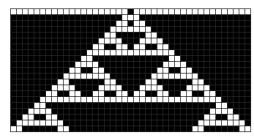
But for the cellular automata that I have discussed in this section, it so happens that the computations they perform can also conveniently be described in terms of traditional mathematical notions.

And this turns out to be possible for some of the cellular automata that I discussed earlier in this book. Thus, for example, as shown below, rule 94 can effectively be described as enumerating even numbers. Similarly, rule 62 can be thought of as enumerating numbers that are multiples of 3, while rule 190 enumerates numbers that are multiples of 4. And if one looks down the center column of the pattern it produces, rule 129 can be thought of as enumerating numbers that are powers of 2.


rule 94


rule 62

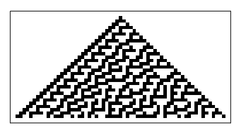

rule 190


rule 129

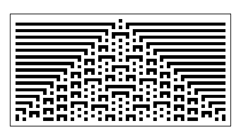
Examples of simple cellular automata whose evolution corresponds to computations that can easily be described in traditional mathematical terms. In analogy to the previous page, the positions of white cells at the bottom of the rule 94 picture correspond to even numbers, on the left in rule 62 to multiples of 3, in rule 190 to multiples of 4, and in the center column of rule 129 to powers of 2.

But what kinds of computations are cellular automata like the ones on the right performing? If we compare the patterns they produce to the patterns we have seen so far in this section, then immediately we suspect that we cannot describe these computations by anything as simple as saying, for example, that they generate primes.

So how then can we ever expect to describe these computations? Traditional mathematics is not much help, but what we will see is that there are a collection of ideas familiar from practical computing that provide at least the beginnings of the framework that is needed.


rule 30


rule 45


rule 73

Examples of cellular automata that have simple underlying rules but whose overall behavior does not seem to correspond to computations with any kind of simple description in standard mathematical or other terms.