EXCERPTED FROM

STEPHEN WOLFRAM
A NEW KIND OF SCIENCE

SECTION 12.1

Basic Framework

12

The Principle of Computational Equivalence

Basic Framework

Following the discussion of the notion of computation in the previous chapter, I am now ready in this chapter to describe a bold hypothesis that I have developed on the basis of the discoveries in this book, and that I call the Principle of Computational Equivalence.

Among principles in science the Principle of Computational Equivalence is almost unprecedentedly broad—for it applies to essentially any process of any kind, either natural or artificial. And its implications are both broad and deep, addressing a host of longstanding issues not only in science, but also in mathematics, philosophy and elsewhere.

The key unifying idea that has allowed me to formulate the Principle of Computational Equivalence is a simple but immensely powerful one: that all processes, whether they are produced by human effort or occur spontaneously in nature, can be viewed as computations.

In our practical experience with computers, we are mostly concerned with computations that have been set up specifically to perform particular tasks. But as I discussed at the beginning of this book there is nothing fundamental that requires a computation to have any such definite purpose. And as I discussed in the previous chapter the process of evolution of a system like a cellular automaton can for example perfectly well be viewed as a computation, even though in a sense all the computation does is generate the behavior of the system.

But what about processes in nature? Can these also be viewed as computations? Or does the notion of computation somehow apply only to systems with abstract elements like, say, the black and white cells in a cellular automaton?

Before the advent of modern computer applications one might have assumed that it did. But now every day we see computations being done with a vast range of different kinds of data—from numbers to text to images to almost anything else. And what this suggests is that it is possible to think of any process that follows definite rules as being a computation—regardless of the kinds of elements it involves.

So in particular this implies that it should be possible to think of processes in nature as computations. And indeed in the end the only unfamiliar aspect of this is that the rules such processes follow are defined not by some computer program that we as humans construct but rather by the basic laws of nature.

But whatever the details of the rules involved the crucial point is that it is possible to view every process that occurs in nature or elsewhere as a computation. And it is this remarkable uniformity that makes it possible to formulate a principle as broad and powerful as the Principle of Computational Equivalence.

Outline of the Principle

Across all the vastly different processes that we see in nature and in systems that we construct one might at first think that there could be very little in common. But the idea that any process whatsoever can be viewed as a computation immediately provides at least a uniform framework in which to discuss different processes.

And it is by using this framework that the Principle of Computational Equivalence is formulated. For what the principle does is to assert that when viewed in computational terms there is a fundamental equivalence between many different kinds of processes.

There are various ways to state the Principle of Computational Equivalence, but probably the most general is just to say that almost all