EXCERPTED FROM

STEPHEN WOLFRAM
A NEW KIND OF SCIENCE

SECTION 12.11

Implications for Technology

as bizarrely different from human intelligence as many of the simple programs in this book are different from the systems that have traditionally been studied in human mathematics and science.

Implications for Technology

My main purpose in this book has been to build a new kind of basic science. But I expect that in time what I have done will also have many implications for technology. No doubt there will be all sorts of specific applications of particular results and ideas. But in the long run probably the most important consequence will be to introduce a vast new range of systems and processes that can be used for technology.

And indeed one of the things that emerges from this book is that traditional engineering has actually considered only a tiny and quite unrepresentative fraction of all the kinds of systems and processes that are in principle possible.

Presumably the reason—as I have mentioned several times in this book—is that its whole methodology has tended to be based on setting up systems whose behavior is simple enough that almost every aspect of them can always readily be predicted. But doing this has immediately excluded many of the systems that I have studied in this book—or for that matter that occur in nature. And no doubt this is why systems created by engineering have in the past usually ended up looking so much simpler than typical systems in nature.

And with traditional intuition it has normally been assumed that the only way to create systems that show a higher degree of complexity is somehow to build this complexity into their underlying rules.

But one of the central discoveries of this book is that this is not the case, and that in fact it is perfectly possible for systems even with extremely simple underlying rules to produce behavior that has immense complexity—and that looks like what one sees in nature.

And I believe that if one uses such systems it is almost inevitable that a vast amount of new technology will become possible.

There are some places where just the abstract ability to produce complexity from simple rules is already important. One example discussed in Chapter 10 is cryptography. Other examples include all sorts of practical processes in which bias or deadlock can be avoided by using randomness, or in which one wants to generate behavior that is somehow too complex for an adversary to predict.

Being able to produce complexity that is even roughly like what we see in nature also has immediate consequences—say in generating realistic textures and computer graphics or in producing artistic images that we abstractly perceive as having features familiar from nature.

The phenomenon of computational irreducibility implies that to find out what some specific system with complex behavior will do can require explicit simulation that involves an irreducible amount of computational work. But as a practical matter, if one can set up a model that is based on sufficiently simple rules then it becomes more likely that one will be able to make designs and build control devices that work even with some system in nature that shows complex behavior.

So what about computers? Although the components used have shifted from vacuum tubes to semiconductors the fundamental rules by which computers operate have changed very little in half a century.

But what the Principle of Computational Equivalence implies is that there are actually a vast range of very different kinds of rules that all lead to exactly the same computational capabilities—and so can all in principle be used as a basis for making computers.

Traditional intuition suggests that to be able to do sophisticated computations one would inevitably need a system with complicated underlying rules. But what I have shown in this book is that this is not the case, and that in fact even systems with extremely simple rules—like the rule 110 cellular automaton—can often be universal, and thus be capable of doing computations as sophisticated as any other system.

And the fact that the underlying rules can be so simple vastly expands the kinds of components that can realistically be used to implement them. For while it is quite implausible that some simple chemical process could successfully assemble a traditional computer out of atoms, it seems quite plausible that this could be done for something like a rule 110 cellular automaton.

Indeed, it seems likely that a system could be set up in which just one or a few atoms would correspond to a cell in a system like a cellular automaton. And one thing this would mean is that doing computations would then translate almost directly into building actual physical structures out of atoms.

In the past biology—with all its details of DNA, proteins, ribosomes and so on—has provided our only example of programmable construction on an atomic scale. But the discoveries in this book suggest that there are vastly simpler systems that could also be used.

And indeed my guess is that the essential features of all sorts of intricate structures that are seen in living systems can actually be reproduced with remarkably simple rules—making it for example possible to use technology to repair or replace a whole new range of functions of biological tissues and organs.

But given some form of perhaps complex behavior, how can one find rules that will manage to generate it? The traditional engineering approach—if it works at all—will almost inevitably give rules that are in effect at least as complicated as the behavior one is trying to get.

At first biology seems to do better by repeatedly making random modifications to genetic programs, and then applying natural selection. But while this process does quite often yield programs with complex behavior, I argued earlier in this book that it does not usually manage to mold anything but fairly simple aspects of this behavior.

So what then can one do? Occasionally some kind of iterative or directed search may work. But in my experience there are so many different and unexpected things that can happen with simple programs that ultimately the only way to find what one wants is essentially just to do an exhaustive search of all possibilities.

And with computers as they are today one can already often look at trillions of cases—as on page 833. But while this is enough to see a tremendous range of behavior, there is no guarantee that one will in fact run across whatever specific features one is looking for.

Yet in a sense this is a familiar problem. For especially early in their history many branches of technology have ended up searching the natural world for ingredients or systems that serve particular purposes—whether for making light bulb filaments or drugs. And in some sense the only difference here is that in the abstract world of simple programs doing a search becomes much more systematic.

But while traditional engineering has usually ended up finding ways to avoid searches for the limited kinds of systems it considers, the phenomenon of computational irreducibility makes it inevitable that if one considers all possible simple programs then finding particular forms of behavior can require doing searches that involve irreducibly large amounts of computational work.

And in a sense this means that if one tries directly to produce specific pieces of technology one can potentially always get stuck. So in practice a better approach will often be in effect just to do basic science—and much as I have done in this book to try to build up a body of abstract knowledge about how all sorts of simple programs behave.

In chemistry for example one might start by studying the basic science of how all sorts of different substances behave. But having developed a library of results one is then in a position to pick out substances that might be relevant for a specific technological purpose.

And I believe much the same will happen with simple programs. Indeed, in my experience it is remarkable just how often even elementary cellular automata like rule 90 and rule 30 can be applied in one way or another to technological situations.

In general one can think of technology as trying to take systems that exist in nature or elsewhere and harness them to achieve human purposes. But history suggests that it is often difficult even to imagine a purpose without having seen at least something that achieves it.

And indeed a vast quantity of current technology is in the end based on trying to set up our own systems to emulate features that we have noticed exist in ordinary biological or physical systems.

But inevitably we tend to notice only those features that somehow fit into the whole conceptual framework we use. And insofar as that framework is based even implicitly on traditional science it will tend to miss much of what I have discussed in this book.

So in the decades to come, when the science in this book has been absorbed, it is my expectation that it will not only suggest many new ways to achieve existing technological purposes but will also suggest many new purposes that technology can address.