COMPUTATIONAL CREATIONISM

Brian Hayes

A reprint from

American Scientist

the magazine of Sigma Xi, the Scientific Research Society

Volume 87, Number 5
September–October, 1999
pages 392–396

COMPUTATIONAL CREATIONISM

Brian Hayes

The great age of automata, or lifelike machines, began toward the close of the Middle Ages and lasted into the 17th century. The technological marvels of that era were clockwork confections—intricate assemblies of gears, cranks, levers and ratchets. Clocks displayed the phases of the moon and the annual progress of the sun through the zodiac; they had animated figures to strike the hours and entertain onlookers.

From machines that imitate life and the heavens, it is an easy step to the idea that life itself might be a mechanical process and that the stars could be driven by some kind of celestial geartrain. The clockwork universe figures in the thinking of Dante, Galileo, Kepler and Newton. Another exponent of clockwork in the sky was Descartes, who also compared animals to mechanical automata. And Thomas Hobbes wrote: "For seeing life is but a motion of Limbs … why may we not say, that all Automata (Engines that move themselves by springs and wheels as doth a watch) have an artificial life?"

Today, the chronometer's ticking escapement is no longer the epitome of high tech. Brass gears have given way to silicon chips. And as the computer has conquered technology, it has also taken the place of clockwork in metaphor and myth. Novels and films no longer portray us as cogs in a machine we can't control; instead we are bit-players in someone else's virtual reality. At a slightly more serious philosophical level, an ongoing debate asks whether computational processes could account for everything happening in the universe, or whether something more—something nonalgorithmic—is needed. And occasionally the question is asked whether the entire universe might be a vast computer cogitating on The Answer.

The World as Machine

The vision of a cosmic computer has inspired literary and philosophical speculation, but the roots of the idea lie in the everyday practice of computer science. It's the sort of notion that might occur to anyone who spends enough time twiddling bits—especially late at night in a caffeine frenzy. There are two versions of the idea, one belonging to the hardware hacker and the other to the software wizard. The distinction between them is this: In the first case the world is *computing something*; in the second the world is computed *by* something.

The hardware variant springs from the observation that even though computers are complicated and finicky devices, you can build one out of almost anything. The beige box on your desk runs on microelectronic circuits, but in principle all of its functions could be performed by hydraulic or pneumatic or photonic devices. Danny Hillis and his friends built a computer out of Tinker Toys and string. Leonard Adleman performed a computation with strands of DNA in a test tube. Other schemes would compute with enzymes or living bacterial cells or spinning atomic nuclei.

The counterpoint to all this technological diversity is theoretical equivalence. Provided that a machine never runs out of memory and that you're willing to wait long enough for an answer, almost all computers can compute exactly the same set of mathematical functions (and they fail on the same set of uncomputable problems). The proof of equivalence relies on the idea of an emulator: a program that allows one machine to run programs written for another. The usual practice is to show that a given computer can emulate a Turing machine, the theoretical computing device invented by Alan Turing in the 1930s, whose underlying technology is the marking of paper tape.

Should we be surprised that so many kinds of machines can all compute the same things? Forty years ago Eugene Wigner wrote of "the unreasonable effectiveness of mathematics in the natural sciences," asking why differential equations should work so well to describe the physical world. The converse question is just as intriguing. Why do all the resources of the material world lend themselves so readily to computing mathematical functions? Why is it you can pick up just about any spare parts lying about the universe and turn them into logic gates or binary adders?

One answer is that the world *is* a computer. It was designed to have exactly this property. The most celebrated speculation along these lines is found in Douglas Adams's *Hitchhiker's Guide to the Galaxy*. Adams reveals that the planet Earth was constructed as a gigantic computer meant to carry

Brian Hayes is a former editor of American Scientist. *Address: 211 Dacian Avenue, Durham, NC 27701. Internet: bhayes@amsci.org.*

out a five-billion-year inquiry into "the meaning of life, the universe and everything."

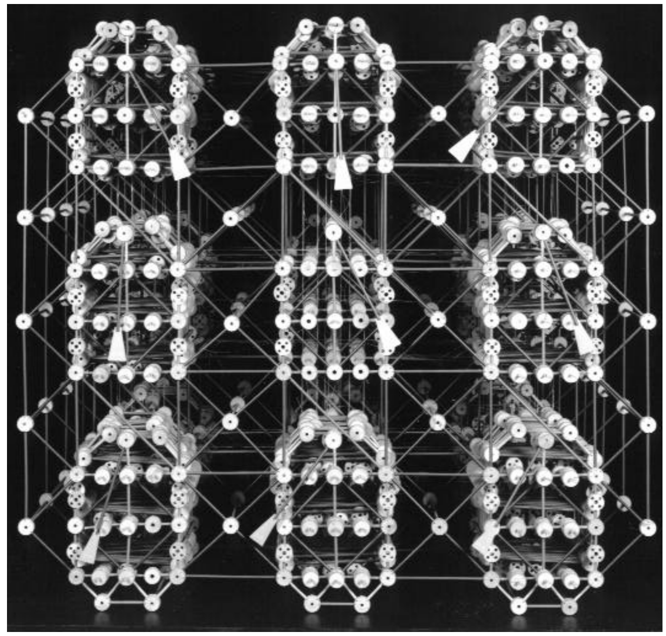
Others imagine computers on an even grander scale, reaching beyond this little wet rock of ours to fill the entire universe. One visionary of the cosmos-as-computer was the late Konrad Zuse, who was also among the earliest of all hardware hackers (he had a digital computer up and running years before ENIAC). Zuse conjectured that the ground fabric of the universe might be a kind of computer called a cellular automaton. This same idea has been pursued with even greater vigor by Edward Fredkin, a free spirit of computer science who led the Information Mechanics Group at MIT in the 1980s.

A cellular automaton is an array of many simple processors arranged in a lattice. Think of a tiled floor with a processor on every tile. Each processor (or cell) has only a finite number of possible states and can communicate with only a finite number of neighboring cells. At each tick of a master clock, every cell chooses its next state according to a fixed "transition rule." The best-known example of a cellular automaton is the Game of Life, invented 30 years ago by John Horton Conway of Princeton University. The cells in Life have two states—alive or dead—and the transition rule simply counts the number of living neighbor cells.

At first glance a cellular automaton doesn't look much like our world. For one thing, our space appears to be continuous: Where are the cells? Fredkin suggests they are simply too fine to see—perhaps as small as the Planck scale, 10^{-33} centimeter. A subtler objection is that our world teems with fast-moving particles, such as electrons and protons whizzing around inside atoms, whereas only signals travel through the lattice of a cellular automaton; the cells are immobile. Here too Fredkin has an answer. A fairly simple transition rule creates packets of information that glide frictionlessly through the cellular automaton like idealized billiard balls, rebounding elastically when they collide. Maybe what we perceive as motion has a similar basis, and elementary particles are made of nothing more substantial than information.

Cellular automata are a natural choice for a computational universe because they require only local communication between nearby processors. There is no need for wires or other long-distance rigging. The deepest laws of nature also seem to be strictly local, making for a good match between physics and computation. These aspects of cellular automata—the dual ideas of "programmable matter" and "computable physics"—have been explored in great detail by Tommaso Toffoli and Norman Margolus, who were both members of the Information Mechanics Group.

In the absence of compelling evidence—and this is a case where we have a compelling absence of evidence—why would anyone choose to believe that the universe is busy churning out calculations? The Douglas Adams fantasy suggests the allure of a hidden purpose. Why are we here? To



The Computer Museum History Center

Figure 1. A computer made of Tinker Toys and fishing line testifies to the idea that almost any kinds of parts can be assembled to perform logical or mathematical calculations. The Tinker Toy computer played tic tac toe. It was built in the 1980s by Danny Hillis, then at MIT and now a Disney Fellow.

compute the meaning of life, the universe and everything. All those events that seem so random and pointless will be explained when the cosmic computer prints out the final answer. (Either that, or the computer crashed ages ago, and we've been waiting all this time for someone to reboot us.)

Fredkin's vision of the universe as cellular automaton is a little different. His computer isn't necessarily searching for bits of wisdom; it may simply be computing its own next state, over and over, with no goal in mind. Yet Fredkin too wonders about invisible undercurrents and mysteries of purpose. He points out that since most of space is empty, most of the cells in the automaton have nothing to do most of the time. He calls this "the problem of the missing workload"; by his estimate, the computing capacity of the universe is greater than needed by a factor of 10^{63}. "Either something else is going on …," he comments, or "God was incompetent on a scale that boggles the mind."

The World as Program

If the hardware fantasy is that you can build a computer out of anything in the world, the software version is that you can make a world inside any computer. Indeed, the sense of personal omnipotence, of creating a domain where you are master, is part of programming's seductive charm. But playing God is dangerous. When you manufacture worlds at will, you become vulnerable to the awful surmise that your own world might be someone else's whim.

As the creator of a computer-simulated world, you decree the laws of nature. If you think gravity should be proportional to the cube of the distance

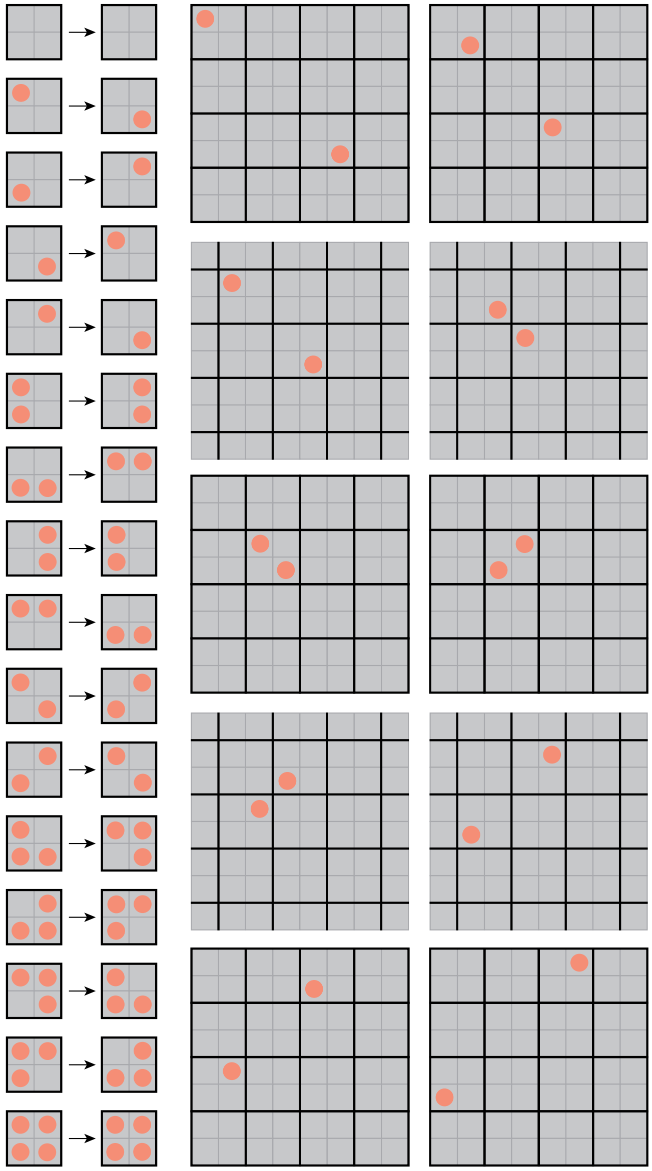


Figure 2. A cellular automaton is one kind of computer that might underlie the physical universe. At left are 16 transition rules for blocks of four cells, with two states per cell. In the panels at right (where time proceeds from left to right and top to bottom), the rules are applied to blocks indicated by heavy lines, which shift position on alternating time steps. The transitions produce a simulacrum of particle motion, even though nothing but information moves from cell to cell. This particular cellular automaton was devised by Norman Margolus of MIT.

rather than the square, then write the equation $F = Gmm / r^3$. Invent water molecules that freeze into eight-pointed snowflakes. Build proteins out of right-handed amino acids. Let the universe expand every Monday through Friday and contract on the weekends.

Does the programmer of a simulated universe have *total* freedom of design? I don't know. If I invent a world, I'm fairly sure I can make the speed of light whatever I please, but it might be harder to tinker with the value of pi, or to make *five*-pointed snowflakes (allowing regular pentagons to tile the plane). Similarly, I don't know how to create a universe where the commutative law of addition fails to hold. But maybe these difficulties just reflect the weakness of my imagination. In any case, even if there *are* limits to the variety of simulated worlds, the programmer clearly has a lot of latitude. And if the laws of simulated nature are so arbitrary, how can we be sure the ground truth of our own world is not the invention of some sleep-deprived programmer?

The fear that the world we know—or think we know—might be nothing but a computer simulation is a nerdish version of a much older idea. In *Through the Looking Glass* we are figments of the Red King's dream; a Taoist parable has a monk dreaming of a butterfly dreaming of a monk. But the computer has put a sharper edge on these musings. Now we have a technology of artificial reality.

The theme turns up frequently in fiction and film. A recent movie by Andy and Larry Wachowski, *The Matrix*, portrays a future in which 1990s urban life is a computer simulation created to mollify an enslaved humanity. (One amusing scene reveals that the phenomenon of *déjà vu* betrays a "glitch in the Matrix," where the illusion momentarily fails.)

The most sophisticated play with these ideas is found in the fantasies of the Polish writer Stanslaw Lem, whose "constructors" Trurl and Klapaucius build a variety of computers, machines and worlds. One of Trurl's disasters is the machine that can make anything starting with the letter *n*—"nimbuses, noodles, nuclei, neutrons, naphtha, noses, nymphs, naiads...." The disaster comes when the skeptical Klapaucius asks the machine to make Nothing, and whole categories of objects begin disappearing from the universe.

In another story Trurl builds a toy kingdom for a deposed tyrant. Klapaucius is appalled:

> "Have I understood you correctly?" he said at last. "You gave that brutal despot, that born slavemaster, that slavering sadist of a painmonger, you gave him a whole civilization to rule and have dominion over forever?... Trurl, how could you have done such a thing?"
>
> "You must be joking!" Trurl exclaimed. "Really, the whole kingdom fits in a box three feet by two by two and a half ... it's only a model...."
>
> "A model of what?"
>
> "What do you mean, of what? Of a civilization, obviously, except that it's a hundred million times smaller."
>
> "And how do you know there aren't civilizations a hundred million times larger than our own? And if there were, would ours then be a model?..."

The story has a happy ending, more or less. Trurl's Lilliputians escape their confinement, overthrow the tyrant and begin playing with nuclear weapons, like any self-respecting civilization.

Hans Moravec of Carnegie Mellon University offers another perspective on the theme in his book *Mind Children*. He imagines a Game of Life where after many ticks of the master clock some of the patterns in the cellular automaton develop consciousness. "The cellular intelligences (let's call them the Cellticks) deduce the cellular nature and the simple transition rule governing their space and its finite extent. They realize that each tick of time destroys some of the original diversity of their space and that gradually their whole universe will run down." So the Cellticks make contact with their creator by spelling out a message on the computer screen. Then the Cellticks and the programmer go off together to explore the programmer's universe, hoping to find another level of reality before this one too runs down.

A Computational Copernican Principle

In most tales of simulated worlds, the tissue of plausibility becomes thinnest at the interface between levels of reality. I can believe (just barely!) in a civilization that exists only as a computer program. Where my suspension of disbelief becomes least willing is in the crossing over between a physical world and an algorithmic one. In movies the leap is often made by putting on a skullcap studded with electrodes or by plugging a cable into your spinal cord. It seems to me there is a fundamental category violation here. I am made of atoms and molecules. How could I enter a world of bits and bytes? (But maybe that's what all simulated creatures say.)

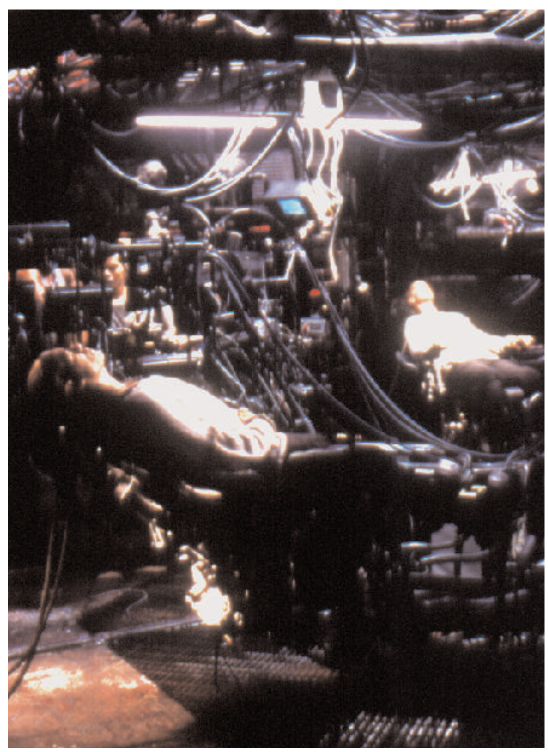
Moravec, in his parable of the Cellticks, handles this issue more carefully than other authors. His programmer never steps into the gridlike space of the cellular automaton. The Celltick program does become an inhabitant of the programmer's world, but not by any kind of magic metamorphosis. The transmigration of souls happens in easy technological stages. First, microphones and television cameras are attached to the Cellticks' computer to give them sensory experience; then the computer is made mobile, so that they can explore on their own.

The most intriguing part of Moravec's fantasy deals with the crucial moment of discovery, when the Cellticks first learn their true ontological status. They take a scientific approach, studying the transition rules that constitute the laws of nature in their universe. "Once in a long while the transition rules are violated, and a cell that should be on goes off, or vice versa …. After recording many such violations, the Cellticks detect correlations between distant regions and theorize that these places may be close together in a larger universe." From this slender clue they learn the structure of the computer that is running the program that creates their world, and they decipher its machine language. We would call this process reverse engineering, but to the Cellticks it is physics.

It seems significant that malfunctions have a role in the Cellticks' cosmological investigation. In a properly functioning computer, a program cannot learn anything about the hardware on which it is running. True, the program might *think* it has learned something. It might go digging through read-only memory and find buried there the telltale markers of an Apple II computer. But the ability of one computer to emulate another makes such digital archeology untrustworthy. The Apple II might be an emulation running on an IBM PC, or a HAL 9000. If the emulators are written correctly, they can reproduce even the most obscure quirks and bugs of the target hardware. Unless you get lucky and spot a glitch in the Matrix, no program will detect the fraud.

Once you begin to take such ideas seriously, the situation goes from bad to worse in a hurry. Consider this: If a simulation is complete enough to have some kind of intelligent entities within it, then those entities could also build computers to simulate worlds, which could include still more computers and simulations of their own. In this tower of simulations, where would our world fit? To answer that question it seems best to invoke a computational Copernican principle. Just as the earth is unlikely to lie at the center of the universe, our level of simulation is unlikely to lie at either the very top or the very bottom of the tower.

This principle can be followed along a curious trail of further arguments. Although we might not be directly aware of any levels of simulation above us, we ought to know about those below us, since they are our own creations. But no such levels exist; we have not (yet) created any artificial civilizations. Thus we seem to be at the very bottom of the tower, which is unlikely, and so it seems safe to assume we are real flesh and blood after all. But this reassuring chain of reasoning has a dark side. If we ever do construct a simulated world rich enough in resources that its inhabitants can create



Photofest

Figure 3. In *The Matrix*, as in other works of science fiction, people living in the physical world are somehow plugged into a world of bits and bytes.

their own simulated worlds, then on that basis alone we might have to conclude that we ourselves are a simulation.

Is the Universe Computable?

One group of scholars would argue that our world *cannot* be a computer simulation because it includes something that is uncomputable, namely the conscious human mind. Three advocates of this view are John Searle, Hubert Dreyfus and Roger Penrose. They marshal quite different arguments in support of their positions, but all three conclude that no algorithmic process could reproduce everything that goes on in the mind. This idea that consciousness guarantees our reality echoes the Cartesian motto "I think, therefore I am."

For those who see no vital difference between brains and computers, the Searle-Dreyfus-Penrose arguments offer no refuge. But perhaps some other computability constraint will intervene. After all, even if it turns out we can simulate a single human mind, it doesn't necessarily follow that we can simulate the entire visible universe.

Writing a program to simulate even a simple physical system—say a few balls on a billiard table—gives you respect for nature's computational abilities. There is so much to keep track of. If you get careless in your collision-detection algorithm, two billiard balls will glide right through each other—a glitch in the Matrix that is sure to be noticed. Performing such a computation for all the atoms in the universe would be truly daunting.

Jürgen Schmidhuber of the Istituto Dalle Molle di Studi sull'Intelligenza Artificiale in Lugano, Switzerland takes up the question of computability in a recent paper titled "A Computer Scientist's View of Life, the Universe and Everything." He concludes that the simplest strategy for simulating the universe might be to compute all possible universes simultaneously. The program for a typical universe would be long and messy, with many tedious special cases. But a trivial metaprogram avoids these complications. It simply enumerates all possible universe-simulating programs in order of increasing length, and executes them simultaneously by interleaving their instructions.

Deus ex Machina

The world seems very solid when you stub a toe, and the suggestion that it might all be a mere pattern of bits appears downright silly. But even an idea that's not taken seriously or literally can have a powerful influence.

The clockwork universe was first of all a theological notion. A clock was thought to imply a clockmaker; and yet, once the clock was wound and set in motion, there was no further need for divine intervention. Thus the religion of the clockwork universe was a cool and inoffensive minimifidianism, with a creator but no presiding ruler. In a similar way, a computational theology might suppose a departed programmer, who clicked on the button marked "Go" and then walked away. But even without a meddlesome programmer on the scene, free will is hard to find in a computing or computed universe. Our actions seem to be ruled by an algorithm whose scope we cannot know.

Perhaps there is a way out. In principle, every detail of a computer's future can be deduced from its present state. Nevertheless, anyone who writes programs has occasionally been surprised by their behavior. Some of the surprises are unpleasant: They are bugs. From another point of view, though, surprises are the whole point of computation. If you could work out in your head everything a program might do, you would have no need to run it on a machine. This idea can be stated more strongly: Some programs are "incompressible," in that no shorter program yields the same result, and there is no faster way of learning what the program does than to run it from start to finish. If our program turns out to be incompressible, we may be acting in ways the programmer never anticipated.

Maybe we can even keep the program and dispense with the programmer. Just as the need for a clockmaker gradually faded from the clockwork universe, perhaps a computational universe could evolve without a computermaker. There is much interest lately in self-organizing systems, emergent computation and evolutionary algorithms. What these buzzwords have in common is the theme of computations done without any need for someone to specify the program in full detail. One of these ideas might allow us to compute our lives away in comfortable anonymity and autonomy.

And a further flight of metaphysical fancy can wipe out the last traces of computational creationism. In the tower of simulations built upon simulations, the ever-nagging question is who built the computer at the top of the tower. But an obvious topological trick will rid us of this inconvenience. Simply wrap the tower around and connect the bottom to the top, forming a vicious circle. In this ring of worlds, we simulate ourselves.

Bibliography

Adams, Douglas. 1979. *The Hitchhiker's Guide to the Galaxy*. New York: Pocket Books.

Fredkin, Edward. 1992. A new cosmogony. http://cvm.msu.edu/~dobrzele/dp/Publications/Fredkin/New-Cosmogony/

Lem, Stanislaw. 1974. *The Cyberiad: Fables for the Cybernetic Age*. Translated from the Polish by Michael Kandel. New York: The Seabury Press.

Mayr, Otto. 1986. *Authority, Liberty, and Automatic Machinery in Early Modern Europe*. Baltimore: The Johns Hopkins University Press.

Moravec, Hans P. 1988. *Mind Children: The Future of Robot and Human Intelligence*. Cambridge: Harvard University Press.

Schmidhuber, Jürgen. 1997. A computer scientist's view of life, the universe and everything. In *Foundations of Computer Science: Potential–Theory–Cognition*, ed. Christian Freksa, Matthias Jantzen and Rüdiger Valk. Berlin, New York: Springer-Verlag.

Toffoli, Tommaso, and Norman Margolus. 1990. Programmable matter: concepts and realization. *Physica D* 47:263—272.

Zuse, Konrad. 1982. The computing universe. *International Journal of Theoretical Physics* 21:589–600.