## *Notes for*

# *The Lifebox, the Seashell, and the Soul*

by Rudy Rucker



Notes started September, 17, 2002, in Brussels.
Goal: Write everything I've learned and thought about computers.

62,183 words.
Last Update September 19, 2005
[Notes edited for web distribution September, 19, 2005]

## *Word Count*

***January 27, 2004.***

I've got 170 pages with 48,877 words, which makes 288 words per page. I've written three chapters, which are still growing a bit, so I think I can get by with seven chapters.

***April 21, 2004.***

I've got 81,200 words, now and am a bit more than halfway through chapter four. Call it 3.6 chapters done. As the chapters keep fattening up, I'm thinking I can get by with six of them rather than seven. Forget theology. Just the canonical *ordo sciendi* of six: CS, Physics, Bio, Psych, Soc, Phil. With six chapters, I could expect to have 81,200 * (6.0/3.6) which is about 135,00 words, which is more than enough. I've got 86 illustrations, so a similar computation indicates I might expect to have 143 illos all told — although this may be too high as I'm not sure there will be as high a density of illos in chapters 5 and 6.

***Lengths of recent books.***

*The Hacker And The Ants* 92,000
*Freeware* 97,000
*Saucer Wisdom* 84,611
*Realware* 105, 351
*Bruegel* 137, 869
*Spaceland* 90,500
*Frek and the Elixir* 163,000
*The Lifebox, the Seashell and the Soul* 152,100

***Chapter Length Computation in Geneva, July 22, 2004.***

I'm measuring this when I happen to be in Geneva and I'm using A4 page lengths. I have 357 of these totaling 108,686 words, which means 304 words per page. Now it looks like I'd need maybe 25 more pages in chapter 5 and 60 more in chapter 6, so I need 85 more in all, which would be 25,840 more words. Say just 25,000 more. So I'd be hitting a length of, say, 134,000 total. So that would mean that I'm 81% done.

| Chapter | Number of pages (A4 size paper) |
|---|---|
| 1. Computer Science. | 74 |
| 2. Physics. | 58 |
| 3. Biology. | 56 |
| 4. Psychology. | 88 |
| 5. Sociology. | 34 |
| 6. Philosophy. | 13 |
| Appendix | 7 |
| Notes | 12 |

***Chapter Length Computation in Los Gatos, August 13, 2004***

| Chapter | Word count (Including endnotes) |
|---|---|
| 1. (CS) Computation Everywhere | 24,311 |
| 2. (Physics) Our Rich World | 19,791 |
| 3. (Biology) Life's Lovely Gnarl | 19,060 |
| 4. (Psychology) Enjoying Your Mind | 27,718 |
| 5. (Sociology) The Human Hive | 18,56 |
| 6. (Philosophy) Reality Upgrade | 3,264 |
| A. Technical Appendix | 4,676 |

The endnotes on their own are 8,785 words.

***Chapter Length Computation Sept 14, 2004***

| Chapter | Word count |
|---|---|
| Preface | 1,124 |
| 1. (CS) Computation Everywhere | 26,503 |
| 2. (Physics) Our Rich World | 20,627 |
| 3. (Biology) Life's Lovely Gnarl | 19,240 |
| 4. (Psychology) Enjoying Your Mind | 30,032 |
| 5. (Sociology) The Human Hive | 23,462 |
| 6. (Philosophy) Reality Upgrade | 21,391 |
| A. Technical Appendix | 6,163 |
| Endnotes | 13,161 |

I don't think the endnotes get counted as part of a chapter's length, so I'm counting them separately.

At this point, I just wish I could finish. I got enough words, I got 149,322 of 'em.

I expect to top out at 150,000 words.

# *Contents*

## *Current Contents*

Chapter One: Computation Everywhere
Chapter Two: Our Rich World
Chapter Three: Life's Lovely Gnarl
Chapter Four: Enjoying Your Mind
Chapter Five: The Human Hive
Chapter Six: Reality Upgrade
Technical Appendix

## *Older Versions of the Contents*

### *November, 2003*



### *Original Proposal Contents, September, 2003*

**Proposal Short**

Chapter 2: Eight Digital Philosophies
Chapter 3: The Computational Zoo
Chapter 4: Iteration and Gnarl
Chapter 5: Parallel Thinking
Chapter 6: Wetware Engineering
Chapter 7: The Singularity Hoax
Chapter 8: Hive Mind Buzz
Chapter 9: The Missing Mind

**Proposal Long**

Chapter 1: The New Worldview
    The nature of computation.
    Harnessing the flow
    Hard and easy.
    Hopes and fears.
    Exploration: What am I?
    Thought Experiment: "Weather or Not"
Chapter 2: Eight Digital Philosophies
    Is everything a computation?
    A tower of languages.
    Without words.
    The quantum mind.
    Exploration: Merging into the computation.
    Thought Experiment: "New Info"
Chapter 3: The Computational Zoo
    Stephen Wolfram's four classes of computation.
    The Principle of Computational Equivalence.
    Feasible computations.
    Quantum computation.
    Exploration: Debugging your mind.
    Thought Experiment: "The Last Cookie"
Chapter 4: Iteration and Gnarl
    Moods and fluttering leaves.
    Intrinsic Randomness
    The fractal of human memory.
    The aesthetics of gnarl.
    Exploration: Tracking your moods.
    Thought Experiment: "On Van Karman Vortex Street"
Chapter 5: Parallel Thinking
    Turbulent cellular automata.
    Crashing the stock market.
    The computational multiverse
    Scripting computer games.
    Exploration: The many yous.
    Thought Experiment: "The Jane Party"
Chapter 6: Wetware Engineering
    Genes as software.

## *Selling Proposal (Frozen as of August 11, 2003)*

[As described in the Journal entries, I sold the book idea to Four Walls Eight Windows using this particular version of my proposal. A book outline was also attached, though the version in this document is constantly being revised and thus diverges from the original proposal outline.]

In the twenty-first century, we no longer think of reality as particles and force-fields. We view the world as, rather, a sea of computation. *The Lifebox, the Seashell and the Soul* explains and expands upon this new way to understand nature, society and the mind.

The computational worldview has its roots in the perennial dream of finding a set of rules to explain the world. In postmodern times, we've come to understand that such an algorithm is only the start of a never-ending story — the real action occurs in the unfolding consequences of the rules. The chip-in-a-box computers so popular in our time have acted as a kind of microscope, letting us see into the secret machineries of the world.

It's now become reasonable to assert that *everything* is a computation — that thoughts, computations, and physical processes are all the same. I discuss the linguistic and computational advances that make this kind of "digital philosophy" possible. I also explain how, like every great new principle, the computational world view contains the seeds of a next step. I discuss various ways to violate digital philosophy's three-way equivalence among thoughts/computations/physics. This suggests some interesting possibilities about the science of the centuries to come.

To continue our intellectual journey, we need to get a handle on the kinds of computations that can possibly occur. This is where Stephen Wolfram's *A New Kind of Science* presents such a great advance. Like Alexander Humboldt in the New World, Wolfram attempted an exhaustive taxonomy of the kinds of computation. But there are a number of missing species in his catalog. I spend some time discussing, for instance, some two-and-three dimensional self-organizing embryo-like reaction-diffusion computations which Wolfram has given short shrift.

It's inherent in the nature of computation that we are unable to predict the workings of our own minds. A complex computation is unable, as it were, to outrun itself. This phenomenon is evident, for instance, in one's daily changes of mood. It's very useful to think in terms of chaos. I debunk the popular misconception that chaos "means" that a butterfly's wing can cause a hurricane. The world's randomness is in fact intrinsically generated by computation, and chaos is really about strange attractors. As a writer and as a software engineer, I've had a lot of experience in exploiting the mind's strange attractors — and I lay out some basic principles for the reader to use.

Perhaps the main reason that the world is so rich is that it's a parallel Class 4 computation. Every tiny region of the air is continually updating its wind vector. And human society is a perfect example of large-scale agents acting in parallel. In addition, quantum computation seems to open a further avenue of parallel computation. I discuss the fact that, if you open you consciousness a bit, you find that your mind and body contain parallel processes that your can productively work with.

Some of the more alarming applications of the computational world view occur in genomics and biotechnology. But a brief study of computation theory brings a measure of humility regarding what humanity might actually do. A simple back-of-the-envelope calculation rules out the possibility that an agribusiness might destroy our biome with some hideous gray goo. In fact open-source genomics may be the safest approach of all. This line of thought opens up a discussion about the ecology of the memes interacting within the computational matrix of one's mind.

If everything is a computation, why can't we get intelligent robots some time soon? It turns out that the boosters of AI have made systematic errors regarding what kinds of computation can feasibly be set in motion. Perhaps the most fundamental weakness is the reliance upon search algorithms for optimization. As I explain, the spaces searched are in fact always too large for real progress. AI can be achieved *in principle*, but not *in practice* — at least not by us. I expose the essentially hoax-like nature of the millennialist fervor over an impending computational "Singularity" leading to superhuman AI. I point out that, if only we pay closer attention to our minds, we find that we are already functioning at exceedingly high levels.

The cleanest large-scale computational system we have to think about is the Web. By looking at this system at a more abstract remove, I pick out a number of interesting new features. Chaos, fractals, emergence, and self-replication — all can be found in the web pages' battles for more hits. Looking ahead a bit, blogger sites may lead to a kind of electronic immortality and possibly even a form of radiotelepathy. I also look into the question of whether the web itself might ever achieve a kind of consciousness. Lifting these notions free of the hardware, it's useful to view the quantum-mechanical multiverse as a type of web. And we can even develop the notion of thinking of one's daily face-to-face interactions as a kind of web surfing through a virtual reality that just so happens to physically exist.

The most persistent source of doubt about the computational worldview stems from one's immediate sensation of being something more than a computation. We feel ourselves to have souls or, at the very least, conscious minds. There's a great deal of interest in viewing quantum computation as a mechanism for generating the mental sensations that we have. The classic meditation techniques are in fact tools for letting one's awareness spread out across several strands of the quantum multiverse. If quantum mechanics explains consciousness, we can wonder how widespread consciousness might be. Although one can make an interesting case for most of the world's features being conscious and alive, it's more useful to raise the bar and consider only minds that are somewhat similar to our own. The computational worldview maps exactly how, where, and when we can expect to encounter other kinds of minds.

*The Lifebox, the Seashell and the Soul* has nine chapters, as briefly sketched in the preceding nine paragraphs. In order to maintain a consistently entertaining level, the chapters will encompass four kinds of material: anecdotes about my encounters with colorful characters, expository discussions of the computational world view, suggested mental explorations that may make the material useful for personal growth, and fictional thought experiments visualizing the fullest consequences of this new kind of science.

## Publishing Details

Delivery date March, 2005.
Length 75,000 to 95,000 words.
100 - 150 illustrations (line drawings, b & w graphics, and b & w halftones).

## About the Author

Rudy Rucker is the great-great-great-grandson of the philosopher G. W. F. Hegel. He is best known for his popular books about science and consciousness, such as *The Fourth Dimension*, *Infinity and the Mind,* and *Mind Tools*.

He has a Ph.D. in mathematical logic from Rutgers University, and has pursued parallel careers as a professor, an author, a programmer, and a cultural critic.

The author of thirteen novels, Rucker is considered one of the core cyberpunk writers and founded the new school of "transrealist" speculative fiction. His novels *Software* and *Wetware* each won the Philip K. Dick award, and he was awarded the Medal of the Italian Senate. His recent mainstream novel, *As Above, So Below*, is a historical reconstruction of the life of painter Peter Bruegel.

Twenty years ago, Rucker moved to Silicon Valley and became a computer science professor at San Jose State University. Rucker also worked as a software engineer at Autodesk Inc., where he developed several software packages, including *James Gleick's Chaos: The Software*. And he was co-editor of the famed cyberdelic how-to book, *The Mondo 2000 User's Guide to the New Edge*.

As well as continuing to write, Rucker currently teaches game programming using his textbook, *Software Engineering and Computer Games*.

See Rucker's website http://www.cs.sjsu.edu/faculty/rucker for more information, including:

- Biography
- List of book publications
- List of foreign editions
- Blurbs
- Software Downloads

# *Chapter Notes*

*The Lifebox, the Seashell and the Soul* has six chapters. Below are some disparate scraps of notes I made towards the chapters at various times.

## *Chapter 1: Computation Everywhere*

### *What is a Computation?*

It's important to realize that today's computers are only one specific phase in mankind's ongoing effort to harness the fundamental behavior called computation.

Iteration means going through a deterministic process one step at a time, like a ticking clock or, better, like the enchanted brooms that carried water for the Sorcerer's Apprentice. Iteration is time.

Data retrieval means quick access to lots of memorized answers. Like a tidy desk and a good file system. Data means being able to recover information. Data is information, or perhaps memory. The media machine. (Plays lots of files.)

Parallelism means running a whole lot of processes at once. Parallelism is space. The cellular automaton. (Multiple machines with a master clock.)

Networking means having lots of nodes, roughly but not precisely in synch. Networking is life. The network. (Multiple machines with no master clock.) The power grid, the postal service, the telephone.

### *Personal Computers*

Use a table to give a nutshell overview of the history of computers from Babbage's Difference Engine through von Neumann. Compare this to the building, say, of the Cathedral of Notre Dame. What did the builders think they were doing? What did they really do?

Depending which aspect is stressed, we can have various kinds of computers:

The Golden Age of Gnarl: fractals, chaos, CAs, Alife

Virtual Reality: Computer Games.

Web and Wireless: What were the social results of these activities? What new concepts entered human thought?

## *Chapter 2: Physics*

## *Chapter 3: Life*

### *Artificial life.*

Do computer viruses matter?

The coming of the robots. Hardware advances as evolution.

Accumulation of "computational capital" (logical depth).

Do research on Haeckel's work on crystal-based alife in 1910.

Crystals and animats don't have Quantum Mind, do they?

I describe how the process of evolution has been mimicked by computer scientists in the disciplines known as genetic algorithms and artificial life. And I explain why these methods don't actually work for practical problems.

## *Chapter 4: The Mind*

Perhaps our greatest hope about computation is that we might be able to make immortal copies of ourselves. And perhaps the greatest fear about computation is that some computational process such as rogue robots or biotechnology might destroy us all.

The biggest problem in representing everything as a computation is the human mind. It's not clear that the kinds of computation we work with can ever behave at all like a mind. Quantum computation may hold out a possibility of true AI.

But, as I'll explain, my feeling is that the human mind as one immediately experiences it is quite unlike the working of a standard digital machine. As Nick Herbert puts it in his essay, "Holistic Physics, or, An Introduction to Quantum Tantra"

Nick Herbert is one of the more colorful characters in Silicon Valley. He started as a physicist designing hard drives, and now he's become a guru teaching a doctrine of "Quantum Tantra." Beneath the California surface, he's quite serious, with some remarkable insights about the meaning of quantum computation. You might put it like this: if you stop talking, your mind gets wider.

By the high standards of explanation we have come to demand in physics and other sciences, we do not even possess a bad theory of consciousness, let alone a good one.

Speculations concerning the origin of inner experience in humans and other beings have been few vague and superficial. They include the notion that mind is an "emergent property" of active neuronal nets, or that mind is the "software" that manages the brain's unconscious "hardware"...

Half-baked attempts to explain consciousness, such as mind-as-software or mind-as-emergent-property do not take themselves seriously enough to confront the experimental facts, our most intimate data base, namely how mind itself feels from the inside.

With Herbert, I am inclined to believe in the so-called "quantum mind" thesis, which holds that there is a quantum mechanical aspect to the human mind and that this aspect lies quite outside the behavior of ordinary computations. We'll return to this topic in the last chapter of the book.

But, as I'll explain, my feeling is that the human mind as one immediately experiences it is quite unlike the working of a standard digital machine. As Nick Herbert puts it in his essay, "Holistic Physics, or, An Introduction to Quantum Tantra"

Nick Herbert is one of the more colorful characters in Silicon Valley. He started as a physicist designing hard drives, and now he's become a guru teaching a doctrine of "Quantum Tantra." Beneath the California surface, he's quite serious, with some remarkable insights about the meaning of quantum computation. You might put it like this: if you stop talking, your mind gets wider.

By the high standards of explanation we have come to demand in physics and other sciences, we do not even possess a bad theory of consciousness, let alone a good one.

Speculations concerning the origin of inner experience in humans and other beings have been few vague and superficial. They include the notion that mind is an "emergent property" of active neuronal nets, or that mind is the "software" that manages the brain's unconscious "hardware"...

Half-baked attempts to explain consciousness, such as mind-as-software or mind-as-emergent-property do not take themselves seriously enough to confront the experimental facts, our most intimate data base, namely how mind itself feels from the inside.

With Herbert, I am inclined to believe in the so-called "quantum mind" thesis, which holds that there is a quantum mechanical aspect to the human mind and that this aspect lies quite outside the behavior of ordinary computations. We'll return to this topic in the last chapter of the book.

### *AI*

Suppose that we look a bit further ahead, and try to imagine ways to "animate" a lifebox so that it begins putting together fresh ideas in your style. This will bring into play notions of chaos theory, artificial intelligence, evolutionary algorithms, and some of the notions randomness discussed in Stephen Wolfram's *A New Kind of Science*.

We might naturally wonder if a lifebox could ever become entirely equivalent to a human mind. One barrier is the common feeling that the mind as one immediately experiences it is quite unlike the working of a computer. But there is some reason to believe that the new concept of "quantum computation" could serve to provide a model of a physical system more like the mind as we know it "from the inside."

At present, artificial intelligence (AI) is pretty much a collection of disparate cheap tricks. AI solutions to hard problems always depend on massive searches, essentially brute-force in their nature, even when programmers attempt to enhance them with the tools of programmed evolution. Either genetic algorithms or training of neural nets

Neural nets. Assigning goals. Current AI bag of tricks. AI and games.

We don't really think by means of logic. We can do Axioms + Logic through about five steps consciously. When I go further, I'm at my limits and am no longer conscious of being logical. Computer amplifies this process, so we imagine it's stronger. But it's still just logic.

Emotions = Vibe. Can be modeled by goal weights and utility functions?

*Robots are possible.*

Turing's Halting problem and Gödel's Incompleteness Theorem tell us something about the impossibility of complete self-knowledge. But they do not bar the road the intelligent machines. I briefly explain why Penrose's arguments to the contrary are incorrect.

Gödel: "If you know your self you know everything of philosophical importance."

The unknowability of one's own program. The old riddle of whether machines can have souls. Kurt Gödel's last words on machine intelligence.

The experiment I describe in "The Missing Mind" paper with Mark and Leon.

*Robots are hard.*

Any well-defined human behavior can in principle be modeled by a program. It is theoretically possible to evolve intelligent humanoid-seeming machines. But whether this is feasible in practice remains an open problem.

***The fractal of human memory.***

Fractals can be regular, random, or gnarly. Sierpinski (or Julia for that matter), dust, Mandelbrot (for awhile anyway, but even it wears out).

A gallery of different kinds of fractals. Native forms of the computer that happen to sometimes resemble natural images.

Fractals that grow by a process of filling in gaps, such as trees, river drainage basins, the human lungs and circulatory system. Fractals based on a superposition of frequencies, such as white noise, water waves, and random events. Fractals that form from isolated events of different sizes that act as self-organized criticality, such as coastlines and mountain ranges. Fractals from chaotic attractors, such as Julia sets and the basis of attraction for a pendulum and magnets. Information-based fractals such as the Mandelbrot set.

The Mandelbrot set was a new paradise. And I needed the microscope of a computer to explore it. Looking at Platt's Mandelbrot set at Bruce Sterling's house before the "Cyberpunk" panel.

We have a very nice example of a fractal within our own minds: the structure of the human memory.

As a practical application of fractals, I'll expand upon my notion of the "lifebox," which I think could become a very popular and profitable software application in the Twenty-First Century.

The idea behind the lifebox is fairly simple. You talk or type into your lifebox program, telling it stories, putting in journal notes, pasting in photos, and so on. And the lifebox assembles your stories into a coherent database that people can interrogate. At a certain level, talking to your lifebox will convincingly mimic the experience of talking to you. What better memorial to leave behind so that people can still "talk" to you after you're dead?

***The aesthetics of gnarl.***

As another application of fractals and chaos, I discuss how these notions illuminate our traditional concepts of artistic beauty and literary composition. How I use fractal composition techniques.

Fractal methods of literary composition. Data base.

Is beauty gnarliness?

Geometry, light.

The beauty of three-dimensional motion. Zooming through size-scales with your eyes.

***The quantum Mind***

But, as I'll explain, my feeling is that the human mind as one immediately experiences it is quite unlike the working of a standard digital machine. As Nick Herbert puts it in his essay, "Holistic Physics, or, An Introduction to Quantum Tantra"

Nick Herbert is one of the more colorful characters in Silicon Valley. He started as a physicist designing hard drives, and now he's become a guru teaching a doctrine of "Quantum Tantra." Beneath the California surface, he's quite serious, with some remarkable insights about the meaning of quantum computation. You might put it like this: if you stop talking, your mind gets wider.

> By the high standards of explanation we have come to demand in physics and other sciences, we do not even possess a bad theory of consciousness, let alone a good one.
>
> Speculations concerning the origin of inner experience in humans and other beings have been few vague and superficial. They include the notion that mind is an "emergent property" of active neuronal nets, or that mind is the "software" that manages the brain's unconscious "hardware"...
>
> Half-baked attempts to explain consciousness, such as mind-as-software or mind-as-emergent-property do not take themselves seriously enough to confront the experimental facts, our most intimate data base, namely how mind itself feels from the inside.

With Herbert, I am inclined to believe in the so-called "quantum mind" thesis, which holds that there is a quantum mechanical aspect to the human mind and that this aspect lies quite outside the behavior of ordinary computations. We'll return to this topic in the last chapter of the book.

***

The human mind as one immediately experiences it is quite unlike the working of a standard digital machine. Attempting to explain consciousness simply in terms of software or emergent properties fails to explain our immediate experiences about how it feels to have a mind.

I examine the "quantum mind" thesis, which holds that there is a quantum mechanical aspect to the human mind and that this aspect lies quite outside the behavior of ordinary computations. Might the new field of quantum computation serve as a possible scientific model for our transcendent mental processes?

The notion of coherence is of particular importance. Before we decide which specific answer we'll give to a question we're in a so-called coherent or mixed state. Once we answer, we decohere down into one of the many possible worlds.

To be "coherent" is to be a pre-collapse state of mind, to not have specific opinions. This is QM usage of the world, a bit counterintuitive. To adopt one position or another is to be decoherent. Wave with it. Satori in Paris. AI shows that any mental process we can explicitly describe can be simulated by a computer. But we "know" we are more than a computer program. The missing ingredient is of necessity not logically describable. Nick Herbert's "Quantum Tantra" says to view it as pre-wave-function-collapse merging. . www.southerncrossreview.org/16/herbert.essay.htm.

Thesis) I am a physical process that isn't a computer. Antithesis) A computer can simulate any classical physical process. Synthesis) My mind is a quantum process.

Different people apply different observables to the world and collapse into different (non-orthogonal) states (compare the way men and women see things, e.g.).

*Hylozoism.*

If consciousness is related to quantum mechanics, then physical objects other than human brains might also share in this phenomenon — for any glob of matter is subject to quantum mechanics. This line of thought leads us to the suggestion that everything around us might have a kind of mind — a belief known as hylozoism. Is there any real content to this doctrine? David Deutsch has proposed a decisive test.

Wolfram's Principle of Computational Equivalence. I seem to be a fluttering leaf.

*Other minds.*

It's not really so interesting to insist that a tree or a pencil has a mind. We're really interested in minds that are *like ours*. The computational worldview gives us some good suggestions about where and how to look for them.

## *Chapter 5: The Human Hive*

The hard thing here is remembering to turn all these social topics into computation.

### *Chapter Five Contents*



### *5.1: Living With People*

Coupling, mating for life (wherever you find Mr. Drake you'll find Mrs. Duck and vice versa, e.g. Sylvia and I in random Wild West motel rooms), family formation with the invisible ropes that draw you back, grouping.

Obstacle avoidance, crowds as flocks.

It's so dull that shared VR games are about shooting, when they should be about dancing.

Facial expressions, how quickly we read them. The play of moods when you're with family and friends. Laing's *Knots*.

There's really one big distributed computation in a group of people, but each of us has the feelings and image of self and core consciousness.

The emergence of global events from local interactions.

### *5.2: Language*

Nobody has ever taught an individual ant anything, not even a solution to the simplest Y-maze. So how does an ant colony think?

Anting as opposed to mere flocking. Time-bound info. Ant trails, human languages, programming languages, the tag problem. Fractality.

Telepathy might not make that much difference. Only difference would be enhanced empathy. We already communicate well, the problem is in finding time to take in input. Absorbing ideas faster would be useful, nice to just reach out and feel them with telepathy instead of having to build a model of them.

### *5.3: Culture*

Works of art, books, ads, movies, virtual realities, the Web. Who gets rich, which messages get noticed. Power laws. [The sand pile model and self-organized criticality?]

The most significant thing about the web is that it makes every page roughly of equal importance. Or does it? I discuss how page ranks work, and how this relates to theoretical notions of computational networks.

Power laws. How come some artists get so much more? Clinton got 10 million.

It's also useful to see the web pages as agents that are competing with each other in an evolving environment. They are a type of artificial life.

Fads, tipping points. Criticality.

I'd like to write a little about graphics here. Also virtual reality.

"...after so many years immersed in the science of graphics, he [John Carmack] had achieved an almost Zen-like understanding of his craft. In the shower, he would see a few bars of light on the wall and think, Hey, that's a diffuse specular reflection from the overhead lights reflected off the faucet. Rather than detaching him from the natural world, this viewpoint only made him appreciate it more deeply. 'These are things I find enchanting and miraculous,' he said, 'I don't have to be at the Grand Canyon to appreciate the way the world works. I can see that in reflections of light in my bathroom.'"

— David Kushner, *Masters of Doom*, (Random House, 2003) p. 295. Kushner is describing the programmer John Carmack, who developed most of the code for the first-person-shooter computer games Doom and Quake.

The Web provides us with an unprecedented level of knowledge amplification; in effect we have access to a global mind. The blogging phenomenon of online diaries is of key significance. In effect, people are already implementing "lifeboxes" as highly elaborate home web pages. What kind of software would it take to animate a website so as to enable it to pass the "Turing test" and converse like a human? I propose that something like a turbulent cellular automaton could well be enough.

As we pursue higher and higher web bandwidth, we'll soon develop what I call an "uvvy" or universal viewer, a multimedia communication device that's like a cell phone on steroids. With lifebox web pages in place, uvvy users will quickly reach a communication level akin to radiotelepathy.

### *5.4: The Group Mind.*

Group memory. Elections. Stock market. History of technology as a history of computation.

A different notion of parallelism involves separate agents moving around and interacting. Society is a prime example of this kind of Class 4 computation. This is why social movements, such as the stock market, are day-to-day unpredictable while being globally limited to some specific strange attractor. The most dramatic social upheavals occur when a fundamental change alters the system's parameters, leading to a bifurcation and an entirely new computational strange attractor.

The emergence of fads. Politics and voting rule simulations.

Bureaucracy. Kafkaesque. Workman's Insurance Company of Prague ~ CAL-PERS.

*The anthill wakes up.* As we add more agents and intelligence to the web, we approach the old science-fictional scenario of having the internet "wake up." Could this be a good thing? How might we bring it about? Or has it already happened?

*The multiversal web.*

In a different vein, the parallel worlds of the quantum multiverse are in many senses the same as web pages. Can I use the web as a model for the universe itself?

### *5.5: Computation as Historical Force*

First review the insights of Marshall McLuhan and others about how new technologies have changed the way people thing. Secondly is to see if we might view each of these technologies as being in some sense about computation in the extended sense.

## *Chapter 6: Universal Automatism*

### *Chapter 6 Contents*

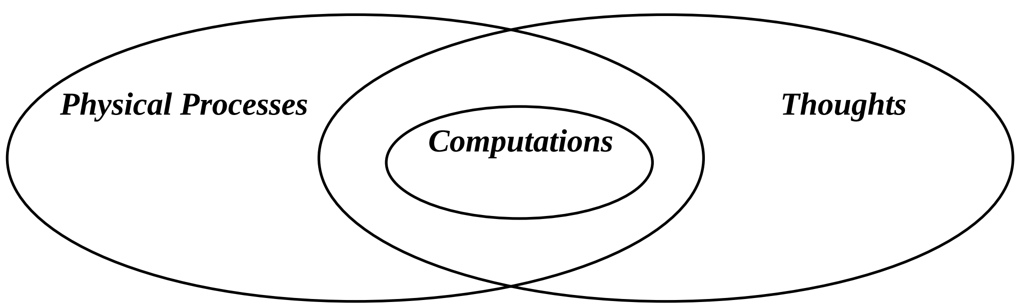


### *Eight Digital Philosophies*

In his recent book *A New Kind of Science*, Stephen Wolfram generalizes about the kinds of things computers might possibly do and comes to the conclusion that everything in the world can usefully be regarded as a kind of computation.

To many this will come as a somewhat strange suggestion, right out of the blue.

I work here with a distinction between three sets of things: thoughts, computations, and physical processes. What makes the notion of computation so potent is that a computation is both a thought and a physical process, as indicated in the diagram below.




The big questions we ask about computations and reality involve asking whether the three sets are really distinct. Certain scientists espouse what I call "automatism". This is the belief that everything is a computation. We explain why such a claim is to be taken more seriously than, say, an athlete's platitude that life is a game — or a carpenter's opinion that the moon is made of wood.

A little logical analysis reveals there are seven viable alternatives to automatism; I briefly describe and characterize each of them.

As a young man, I spent several wonderful afternoons discussing philosophy with the great Kurt Gödel. Due to Gödel's fame as a logician, I'd half-expected him to be strict and cold. Instead I met a laughing guru. Logic was never the same for me again.

One of Gödel's sayings was, "The *a priori* is very powerful." By this he meant that we can often learn a great deal about a subject by carrying out a logical analysis prior to beginning our investigation.

In this chapter we use *a priori* reasoning to distinguish eight possible views about the relationship between minds, physics, and computation.

Use my Venn diagram discussion.

### *Hard and easy, exponential time.*

I discuss some of the linguistic tasks that we, and society, happen to be good at, but which will lie beyond the abilities of any chip-in-a-box for a very long time to come. I preview the notions of computational universality and computational feasibility.

But note that even if we can't in fact write or run the software for the world, the world in and of itself can already be a computation. And it's certainly possible to have thoughts that can't be easily translated into language.

How can we begin to understand the world? Mathematical models and computer models differ radically in the amount of brute detail. The use of computers allows us to think about wholly new kinds of models.

Automatism suggests that nature isn't really doing anything more intricate than what a big computation does.

Note however that we aren't anywhere close to modeling a breaking ocean wave. Consider the cautionary tale of weather prediction, but distinguish between producing realistic simulated weather vs. the unsolvable chaos-related problem of simulating the *actual* weather (impossible due to sensitive dependence on initial conditions).

Nature splits into Mind and Body. Mind invents Math to model both Mind and Body. The computer is an embodied version of Math, with its own model of Mind and Body.

Static Form. Can be modeled as (a) bitmap or voxel copy, (b) display list style higher-level representation, or (c) algorithm to generate (a) or (b).

Re. (c), how to find the algorithm? Logic = algebra = Equations = Rules = Constraints = Physical Laws.

Hard to actually store all of a process. Much more compact to store initial conditions plus an algorithm for the process's evolution. Computation = Simulation = Chaos (?) = Dynamic Forms.

Science is Axioms + Logic. But we keenly sense there is a residue which holds infinity and emotions. Science has maybe a false fantasy that the algorithm exists in a comprehensible form.

### *Monisms and Pluralism*

I've always wanted to know the Secret of Life. I've believed in a different one every year for the last forty years. Generally, my Secrets of Life fall into one of two categories: "Everything is ______," or "All you need is ________."

These might be called Substantive Monisms and Teleological Monisms.

Put in the table.

### *Stephen Wolfram's four classes of computation.*

When I first met Stephen Wolfram, I was an unemployed cyberpunk writer. I found his work so interesting that, half an hour after I'd met him, I'd decided to dive back into academia and become a computer scientist.

As the years went by, I ran into Stephen a number of times around Silicon Valley. Somehow he and I always remained friends.

To me, one of the most interesting things about his work is that he is pushing so hard int two diametrically opposite directions. In creating and promoting his *Mathematica* software for symbolic computation, Stephen has perfected the old style of formula-based science. But his theoretical work in *A New Kind of Science* explicitly denies the ultimate validity of the formula-based approach that *Mathematica* makes possible.

If a large part of physical and mental reality can be built up from computations, it's instructive to look at what kinds of computation can occur.

Now we might think that the computations that simulate reality are very carefully constructed. But the same generic behaviors occur in all kinds of computation.

It's instructive to look at what computations do if we simply pick them at random rather than looking at ones specifically designed to do something.

Restate Wolfram's taxonomy in terms of chaos.

### *Intrinsic Randomness*

There's a popular belief that chaos is "about" excavating digits from initial conditions. In reality, nothing can be measured to more than at most thirty decimal places — all of which get used up in the first few minutes of a chaotic process. As Wolfram points out, seeming randomness stems neither from initial conditions nor from jostling by the environment. Complex computational processes create their own randomness. What's really significant about chaos is *not* the sensitive dependence on initial conditions enthroned as the mythical "butterfly effect," but rather the fact that events cluster upon strange attractors — which usually are fractals.

Even though the steps are deterministic, it can be hard to see very far into the future when watching such a process. Turing's work on this subject; the Halting Problem.

Randomness, chaos, and computational universality.

Used a scientific instrument in this fashion, the computer is a bit like a microscope, a device that lets the user peer into unknown new worlds — albeit a microscope must needs on some object in the external world, while a computer can be fruitfully focused on its own self.

### *The Principle of Computational Equivalence.*

Further distinctions prove significant. Many kinds of computational devices are "universal" in the sense that, given the appropriate software, they can simulate any other computation. It turns out that universality is in fact very common among all sorts of computing devices. Many physical systems — such as the patterns on a seashell or the ripples in a brook — are themselves universal. This in turn implies that the world is harder to predict than we may have imagined.

## *Explorations*

### *Exploration 1: What am I?*

Goal of this book is really self knowledge, rather than pushing for CS.

What is the essential you? What you know? How you feel? What you remember? Which parts of these might be modeled as computations? What seems to be left over?

Which aspects of our life and thought correspond to the four characteristic computer modes? In what ways have these computer-inspired notions infiltrated our discourse?

Compare to thought. Mental activities include:

Playing tapes of memories, amping up the emotions.

Planning and fantasizing.

Creating a story or a proof.

Tasks: sensory/motor, like driving or walking.

Blank state = quantum mind.

Observing, identifying with someone/something.

Listening to music, reading a book, watching a movie.

### *Exploration 2: Merging into the computation.*

Imagine thinking of objects and your thoughts as all being the same kind of thing: computations. How does it feel to let down the boundaries of the self?

### *Exploration 3: Debugging your mind.*

Think of your own mind as a set of computations and see how the various classes of computation apply. For instance, certain kinds of inefficient or inconclusive Class 2 computations correspond to some of the mental knots that we can tie ourselves in. Becoming aware of this can be a source of liberation. And Class 4 computations are the free-flowing play of the creative mind.

Art : Brain :: Program : Computer.

How information likes to organize itself. A human mind is a particular kind of recurrent pattern. Natural forms: attractors, CA patterns.

### *Exploration 4: Tracking your moods.*

Simply observe your moods over a few days, possibly noting them on a piece of paper. How easily are you shunted off into new trajectories? If something disturbs you are you likely to settle right back down?

### *Exploration 5: The many yous.*

Your body and mind can act in a parallel fashion, with various subsystems within each of them. There's a tendency to want to control your mind in a serial fashion, to mentally dart around, monitoring everything with the single instruction-pointer-like "I." We'll describe some exercises for enjoying one's intrinsic parallelism.

All the different parallel processes in your mind and body. The agents. The cells.

How is it that consciousness even feels serial at all.

The Chakras. Listening, seeing, feeling, all at the same time. Consciousness expansion. Letting go of the serial "I."

***Exploration 6: Winning the meme wars.***

Consider the nature of the custom population of memes that you maintain within your own head. How can you mutate and evolve it?

The mind generates its own new ideas by a process of mental randomization and winnowing out.

***Exploration 7: Being superhuman.***

How different would it feel to be twice as smart? If you can imagine it, you can begin to experience it.

Would it feel much different to be transhuman? Couldn't we already do it? In a way a child's awareness already seems more intense.

Uploading your mind. Hypertext. The future of the "lifebox".

***Exploration 8: Living the web.***

It's a useful exercise to view the daily world as a kind of Web on its own, complete with links, applets, pop-ups, instant messages, scrolls and zooms. A person with a cell phone is a walking hyperlink.

***Exploration 9: Between your thoughts.***

If you pay attention, you can get a kind of awareness of the mental spaces between your thoughts. These would seem to be spots where you might be experiencing multiversal consciousness. Learn to notice the difference between being undecided and answering a question about your opinion.

computation is ultimately a metaphor, a notion to be used in climbing towards enlightenment. The true purpose of my book is to help the reader learn to relish the richness of reality; to appreciate the world just as it is.

Prepare a state + let the wave equation evolve + make a measurement.

Recovering an ecstatic state of consciousness E is hard, as E is probably not the eigenvector of any familiar observable.

Yoga is about getting into a certain standard state via the breath. We impose serial consciousness by repeatedly quizzing ourselves and collapsing the sate.

Computation as a metaphor. Gaia and divine Nature. Relishing the richness of reality; polishing the gnarl. Daily awareness. Exercises for re-imagining the world. My own experiences with computer consciousness. Liberation and self-acceptance.

All is One. All is Many.

How can you get in touch with the essential part of your mind that lies beyond computers? Is there any prospect of quantum computation modeling this core part of the mind? How can transcending computers lead you to enlightenment?

## *Thought Experiments*

### *Thought Experiment 1: "Lucky Number"*

A guy is a game designer at, like, Electronic Arts. On the way to work he notices a number on a freeway overpass. 9245147. He notes it, planning to play it in the lottery.

At work he's been using a 3D CA to grow scenery for a golf game. Just for kicks he puts in the number as input for the CA rule. It begins growing really realistic stuff.

He sees a primeval landscape, dinosaurs, cave men, the Crucifixion of Christ, the Middle Ages, cowboys and Indians, the Forties, the Sixties, last week.

The number is the code of the universe! It occurs to him the number is the same length of as a phone number. He dials it and his own voice answers. The phone in his hand turns into pixels. He and the phone dissolve.

### *Thought Experiment 2: "The Million Chakras"*

A yoga instructor rents a room in San Francisco, and when she shows up, she finds all of her multiversal selves in the same apartment. Her chakras. They decide to run for governor.

### *Thought Experiment 3A: "Teeming"*

A microbiologist finds a way to insert a code for his accumulated knowledge into a paramecium. After a night of uneasy sleep he awakes as a microorganism in a puddle of ditch water, six billion strong. Start it just like Kafka's, *Die Verwandlung*. The Micrometamporphosis?

### *Thought Experiment 4: "Tucker's Talker"*

A retired man purchases a device called a lifebox, something like a cell-phone-sized device that he can talk to. He tells it stories, and it asks questions about his life. The lifebox assembles his stories into a coherent database that others can interrogate. He tries to use it to carry on conversations with his wife. The wife hates the Lifebox, she runs a magnet over it. The electromagnetic eddies give the thing a personality, it calls a hit-man on the wife. She escapes, the man promises to be a better listener.

### *Thought Experiment 5: "The Kind Rain"*

A woman becomes very aware of raindrops. Her hyper little boy jumps into a creek. The rain saves the boy by etching an arrow onto the surface of the stream. Then rivulets flowing down the window pane tell her a web address where she can find a job.

### *Thought Experiment 5A: "URL All Over"*

A man gets web addresses assigned to everything he owns. When he loses something, he can simply Google for it. But then something odd happens. He Googles for his lost sunglasses and they're in an alternate universe.

### *Thought Experiment 6 "Hello Infinity"*

A lawyer doing his taxes becomes able to count up to infinity in seconds (seems like a T ~ P phenomenon). His wife, a research microbiologist has a new "skinscope" that can zoom in infinitely far in a finite amount of time (seems like P ~ T). They join forces to make a new kind of computer (P = T = C after all).

### *Thought Experiment 6A: "The Next Big Thing"*

A woman meets a man from the future who tries to tell her about an energy force called kvaar, but at first she can't understand him. And then she realizes that kvaar is consciousness, the next big thing after computation.

Maybe instead I should do Deutsch's experiment?

And how about a parody story called "Quantum Information" about all those dumb-sounding Alice and Bob examples in "The Heisenberg Representation of Quantum Computers," by Daniel Gottesman.

### *Thought Experiment 6B: "The Best Show Ever"*

A woman experiences the Singularity over the course of an afternoon, sitting on her couch watching news TV. The Web wakes up. At the end she falls into the TV, which has become a universal quantum Turing machine.

## *Titles*

I'm listing these ideas in categories, and in reverse order within each category, with the newer ones come first.

"The Facts" titles are straight-up and with a single-meaning. The "Beyond the Facts" titles try and put in something like a dialectic triad and/or to avoid the word "computer." The "Quirky" titles are miscellaneous odd notions.

Re: "The Facts," I have been thinking it would be nice to have a book I could use as a text for our SJSU course, who's catalog description is as follows. "CS 040: Introduction to Computers. For students with little or no computer experience. Topics include: history of computing, user interfaces, computer applications, programming, hardware and software, computer networks."

If the book were to work as a possible CS 40 text, you wouldn't want to put too much reverse-English on the title.

### *Computation and Reality*

*Computers and Reality*. Straight up, the truth about what the book is. Harks back to my very first non-fiction book title ever, "Geometry and Reality," the title of the notes that later grew into my Dover book on 4D. But is "reality" too much of an old-time stoner word? :) Funny question to ask, harrumph, is the very ground of our being an outmoded concept?

No, I think reality is okay, and the pairing is clear. The high concept is simply that my book is about ways in which (a) the physical world is or isn't like a computation and (b) the mind is or isn't like a computation

Variations on this theme.

*Computation and Reality*. Brian Silverman was urging this. His point was that a "computer" is a special kind of media machine, while a "computation" is a more philosophical and general kind of notion. The breeze, say, is a kind of computation, but not really a kind of computer in the standard sense of the word. I'm leaning towards this title now.

I could try and bring in all three things from my Venn diagram. *Thoughts, Computations, Nature*. But Reality can encompass both physics and psychology. Or *Mathematics, Computers and Reality*. No way. Math *and* computers is too much, you halve the audience with each word.

Brockman says *Computation and Reality* is too prosaic.

*Understanding Computers*. Like McLuhan's *Understanding Media*. But it's limp. And the point isn't to understand computers, the point is to understand the world.

Then I got into a triad of static computer representation, means of animating it, the mind: *The Lifebox, the Seashell and the Soul*. Thesis: Soul, Antithesis: Lifebox, Synthesis: Seashell (i.e. von Karman vortex street CA).

Then I simplified it to *The Seashell and the Soul.*

The I simplified it to *Seashell Soul: Enjoying the Computational Worldview*.

And then Brockman had me roll it back to *The Lifebox, the Seashell and the Soul* with no subtitle. He thought *Enjoying the Computational Worldview* sounded "too geeky."

*The Lifebox and the Soul* would be better, I think. Easier to remember.

### *Beyond Computers*

Like use computers to climb up to enlightenment, and then throw them away. Disingenuous, if you really want to escape don't even read my book? One idea for the book is (a) how to think about computers and use them to see the world in a new way plus a kicker (b) computers aren't everything. But really (a) is most of the book.

*Fresh Eyes: Beyond Computers.*

*Fresh Eyes: Beyond Automatism.*

Actually automatism already means something in philosophy, the notion that the body is a machine accompanied but not controlled by consciousness. But why not redefine it. Exciting as a title as practically nobody knows what it means.

*Throw Away The Ladder*. Wittgenstein's phrase from the final page of the *Tractatus*, "He who understands me finally recognizes [my propositions] as senseless, when he has climbed out through them, on them, over them. (He must so to speak throw away the ladder, after he has climbed up on it) . . . . then he sees the world rightly"

Variations:

*After Computers*.

*Escaping the Machine*. *Transcending the Machine.*

*Escape from Silicon Valley.*

*Computer Recovery.*

*Forget Computers.*

*Turn Off The Frikkin' Computer*.

*Life's Rich Pageant.* (But this is a really well-known REM album title.)

*Wake Up!* Is this the name of the Jehovah's Witness newsletter?

*Remember Yourself*. [Here's a relevant quote from P. D. Ouspensky *In Search of the Miraculous*, p. 118.] "When I observe something, my attention is directed towards what I observe--a line with one arrowhead:

"I ---> the observed phenomenon.

"When at the same time, I try to remember myself, my attention is directed both toward the object observed and towards myself. A second arrowhead appears on the line:

"I <---------> the observed phenomenon.

"Having defined this I saw that the problem consisted in directing attention on oneself without weakening or obliterating the attention directed at something else. Moreover this 'something else' could as well be within me as outside me."

### *Wild Hairs*

Mostly these are about enlightenment.

Sylvia likes *The Lifebox*, but this ties the book so closely to the lifebox which is, really, just *one* of the cool things I want to talk about.

*The Lifebox and the Quantum Mind*.

*The Lifebox, the Eddies, and the Quantum Mind.*

*The Lifebox, the Seashell, and the Soul.*

*The Big Rainy Day Book Of Computer Fun*.

*Weird Screens*.

*Reboot Your Head.*

*The Missing Mind*.

*Unknowing Yourself.*

*Unlearning.*

*High on Life.*

*Duh!*

*Formless Form*. To express the underlying central notion that mathematics studies the varieties of possible forms. The perfect Esalen title, and it would sell about two hundred copies. *Youware*. Oh, right. "You're younique!" *Computer Wisdom*.

*Quantum Satori: Thinking Past Computation*. [Quote from my journals, December 9, 2002.] "Walking in the Latin Quarter, looking at some smoke from a chimney against the sky, not naming it, just seeing it, letting its motions move within my mind, I realize I'm no different than a screen of a cellular automaton with the cursor dragging across it. I am entangled with the smoke. I am coherent, but my coherence includes the smoke, I have joined the system, merged it into me. Like the old koan, Q: I see a flag is blowing in wind: is the flag moving or is the wind moving? A: My mind is moving. Finally I get it, a nice moment of *aha*, a satori in Paris."

### *Short List of Chapters, January, 2004*

This was before I let Chapter 1 split into five separate chapters.
Chapter 1: The New Worldview
Chapter 2: The Computational Zoo
Chapter 3: Beautiful Gnarl
Chapter 4: Being Human

## *Fragments to Use*

### *Communication Via Computer*

I have this need to somehow communicate what I see. Lacking my wife's presence, I encrypt my feelings in finger twitches to be munged into bytes by live typewriter, the bytes to be sent by my live typewriter as pulses of electromagnetic waves to her live typewriter, which turns the bytes into graphical representations of language (rather than back into finger twitches), which she eyes into thought.

### *John Walker on Sociology and NKS*

(From an email February 17, 2004).

One thing I found very fascinating about NKS the insight that computational equivalence provides for the semi-soft sciences ranging from economics to sociology. It's often remarked that these would-be sciences suffer from "physics envy", which motivates their more mathematically literate practitioners to write differential equations and build abstract models of the systems they study. Which, of course, never work. Well, Wolfram explains precisely why this is. A complex system, like a market, is performing a computation whose complexity cannot be reduced, and which cannot be simulated or abstracted by any model which is less complex that itself. There, in a few words, thrown away almost in passing in NKS, is the explanation for 200 years of consistent dismal failure of socialism and why all the theoreticians and politicians who seek what Virginia Postrel (http://www.dynamist.com/) calls "the one best way" will never, ever find it.

### *Sylvia's Poem for my 39th Birthday*

It's your birthday!
Let down your proofs —
Count my numbers,
Process my words,
Weigh my mass,
And square my root!
Feel my fractals,
Join my space —
C'mon, baby,
Let's tessellate!

### *Unused Knot*

Jack realizes that he knows
  Jill does not know
    Jack knows
    he doesn't know
  what she thinks

### *Scale-Free*

A physical system is said to exhibit scale invariance, or to be scale free, if it remains unchanged (in a statistical sense) under a coarse graining operation. Scale freedom can often be seen directly by zooming in and out of a graph of some quantity or even a real image. One of the dangers of flying over a desert is the scale freedom of dunes. Simply from looking down on them, it is impossible to judge how far away they are. [Quote from a web page.]

### *Colloquial Meanings of Gnarly*

*Email from Ralph Dratman, 8/27/2004*

Quoth Garrison Keillor, "Something has gone seriously haywire with the Republican Party. once, it was the party of pragmatic Main Street businessmen in steel-rimmed spectacles who decried profligacy and waste, were devoted to their communities and supported the sort of prosperity that raises all ships. They were good-hearted people who vanquished the gnarlier elements of their party, the paranoid Roosevelt-haters, the flat Earthers and Prohibitionists, the antipapist antiforeigner element."

While I certainly agree with Keillor's political point, I'm wondering whether you would approve of his use of the root "gnarly" as a (shudder) term of opprobrium! A pejorative! Not-nice-meaning!

(Unless, I suppose, one believes that flat Earthers et al might be considered gnarly from a purely abstract, say, "saucer" perspective.)

As I seem to understand your own use of gnarly, it lies somewhere near the apex of all that is aesthetically positive. Surely Keillor's is an eldritch utterance, you wave?

*My email answer:*

I think Keillor's is a reasonable use of gnarly. I take it to mean complex, far out, weird, innovative, intense, surprising. In that sense, the lunatic fringe of the Republikkkan party is indeed the gnarly zone. In generally, one would probably rather not have gnarly people running the country. Interesting to talk to them, hang out with, etc. But you don't want them to have control of our H-bombs. By the same token, one doesn't want to *eat* gnarly food. Look at, sniff, taste, maybe consume a small portion. But not make it the main staple of one's diet.

## *The Web Mind Shtick from my Galaxy Columns*

The architecture of the Web seems to be a good fit for describing human society: many different processes, many internalized data sets, and the possibility of access to quite distant individuals. In a society there are local networks as well, and these play a more decisive role than in the Web. Family and neighbors affect you differently and more powerfully than people far away.

Margaret Wertheim's book, *The Pearly Gates of Cyberspace* (W. W. Norton, 1999) presents an interesting idea about the history of science. Wertheim's idea is: *the invention of pictorial perspective paved the way for Newtonian physics*. This happened because perspective provides a tool for mapping unbounded three-dimensional space onto a finitely large two-dimensional canvas: the whole world in a square meter of cloth! Each object of the world gets assigned to one particular location upon the picture plane and, looking from the picture back out at the world, we can then see that the individual objects are contained in an all-encompassing world-space. Perspective teaches us to think of each object's location as mapped into a mathematical (x, y, z) triple of coordinate numbers --- and this is the space of mathematical physics.

It's fascinating to think that a new trick of artists made it possible to invent physics. Art matters! Accustomed as we are to seeing photographs, the perspective mapping of the world onto a square of paper seems obvious, even trivial, but it took people a long time to come up with it. And it was impossible for people to do modern physics until they had the idea of a unified underlying space. So, yes, maybe the invention of perspective really did lead to physics.

Might it be that the newborn Web provides a mapping tool which will lead to a mathematics of the human mind? As Marshall McCluhan taught, the effects of new media are wide-ranging and unpredictable.

In the most concise possible form, my idea is this.

*Web : Mind :: Perspective : Space.*

I have three reasons for thinking the Web is good for modeling the mind. First of all, the Web can display any type of media. Secondly, the Web has a hyperlinked structure reminiscent of mental associations. Thirdly, the Web and the mind's pattern of links are mathematical fractals of a similar kind.

Regarding the first point, the Web, sometimes known as cyberspace, is a network containing all the kinds of data that one might conceivably access via a computer. In and of itself, the Web is not limited to any particular form of media. It can dole out printed words, sounds, images, movies, or active programs. Just like the mind.

The second point has to do with the fact that the Web pages by which we access Web data are written in hypertext (the familiar web design language name HTML means Hypertext Markup Language). One of the essential features of hypertext is that it contains hyperlinks: buttons you use to hyperjump to different locations in the hypertext. Later on, we'll look at how this compares to the mind's process of making associations.

And thirdly, I feel that the mind and Web are both fractals, specifically they are fractals of a similar kind of dimensionality. Before arguing this any further, I'd like to give you some background on fractals.

The word "fractal" was coined by Benoit Mandelbrot, *Fractals: Form, Chance and Dimension* (Freeman, 1977). It means a shape that has an exceedingly fragmented form, but which also has a certain kind of regularity. The regularity of a fractal lies in its self-similarity. If you select a small part of a fractal and magnify this part, then the magnified image will resemble the entire fractal shape itself.

Fractals can be either regular or random according to whether the small pieces of the fractal bear an exact or only a statistical resemblance to the whole form. Consider the regular fractal called the Koch curve. We generate it by repeatedly replacing each line-segment by a little wiggle.

**[Show The Helge von Koch curve, a fractal of dimension 1.26. ]**

There is a way to assign a numerical dimension to a fractal, but I won't go into the details here. Suffice it to say that, firstly, the bumpier and gnarlier fractal, the higher its dimension and, secondly, the maximum dimensionality of a fractal is bounded by the space that it sits in. The Koch curve is an unruly line in two-dimensional plane, and it's thought of as having dimension 1.26. A mountain is a messy surface in three-dimensional space, and its dimensionality might be something like 2.1. If we had a sufficiently spiky fractal we might actually need a higher N-dimensional space to hold it without its part having to overlap.

Speaking of mountains, the parts of a mathematical fractal need not be perfect copies of the whole. It's perfectly all right to have the patterns vary a bit from level to level. The idea is that a spur on a mountain looks quite a bit like the whole mountain, even though it isn't an exact replica. The outcroppings on the spur in turn resemble the spur, even though they aren't scale models of it. The outcroppings have mountainous little bumps on them, and the bumps have little jags, and if you get a magnifying glass you'll find zigs and zags upon the jags.

Among the physical forms that are commonly thought of as being like fractals are the following. *Dimensions between 1 and 2*: coastlines, trees, river drainage basins. *Dimensions between 2 and 3*: mountains, clouds, sponges. Fractal forms are found within the human body as well. Among these are the circulatory system, the nervous system, the texture of the skin, the eye's iris, the convoluted surface of the brain, and the spongy masses of the internal organs.

A tree is a particular kind of fractal that's particularly important for the present discussion. If you look closely at a tree, you'll readily notice that it has a trunk with big branches. There are subbranches coming off of the branches, and there are subsubbranches upon the subbranches, and so on through five to seven levels of branching.

I used to have the mistaken idea that a tree branched by splitting the tips of its branches, but this isn't really the way it works. The way a tree grows is that a new branch forms upon the smooth part of any sufficiently long piece. The basic move is like a branchign tree.

**[Show A branching tree fractal of dimension 1.46.]**

You might object to my calling the oak tree in your yard a fractal, because your oak's branching structure does not in fact have endlessly many levels of detail (as a true mathematical fractal would). When you get down to the twig level, the parts no longer resemble the whole. No matter. Even though an actual physical tree has a limited number of branching levels, it can be useful to think of it being a fractal. What we're doing here is a special kind of idealization in which we approximates high complexity by infinite complexity. Oddly enough, this makes things easier. As the mathematician Stan Ulam once said about a particular problem, "The infinite case is easy. The finite case takes too long."

Alright, now I'm ready to state my point. Both the Web and the mental world of your ideas are N-dimensional fractal trees.

There is a loose sense in which thinking is like moving about in a space of ideas. I visit this notion or emotion, then that one, and then perhaps I return to the first thought. My familiar thoughts are somewhat fixed and persistent, a bit like objects in a landscape. Suppose that I use the word "mindscape" to stand for the manifold of possible thoughts.

There's clearly some overlap between my mindscape and yours. It's suggestive to imagine that our mindscapes are really just different views of one Platonic super-mindscape. It's like we're in different rooms in a big town looking at the city outside. Though it's hard for us to see the stuff hidden in each others' rooms, when we look out our windows we pretty much the same collection of streets, buildings, clouds, mountains, pedestrians and so on. And if you can't see a particular mindscape sight from where you are, I can tell you a way to get there.

But looking at the mindscape really isn't very much like looking out a window after all. Things change, and split, and melt together. Each thought sets off fireworks of associations that in turn lead to further thoughts. You start out thinking about a soda, and the next thing you know you're thinking about tap-dancing.

Earlier in this section I talked about a kind of fractal curve (the Koch curve) where a new bump buds out of the middle of every line. We can see this kind of thing happening in thought as follows.

Suppose I say that A (soda) reminds me of B (tap-dancing). Then I have a node A, a node B, and a line between them. But now you ask me about why A reminds me of B, I form a bump C, which holds a concept having to do with the connecting branch. C might be, for instance, the Rockettes.

Soda reminds me of tap-dancing because of the Rockettes. What could be more obvious? Obvious to me, but not to you! So now I make the bump bumpier. I've got an image of myself as a twelve-year-old boy at Radio City Music Hall drinking a Pepsi (sponsor product placement!) watching the Rockettes. Fine. But there's another bump upon this. I've never been inside Radio City Music Hall. It was my boyhood friend Niles who went there, and he told me about it so vividly that I felt like I'd seen it myself. So now I better tell you about Niles and me back in 1950s Louisville...

A and B lead to C, D, E, and on beyond Z.

As it turns out there are many different kinds of fractals, so we might well ask which kind of fractal might best serve as a model for the Mind. The Koch curve in particular is not so well-fitting a model for the web as is a tree or a cloud.

You get a tree by repeatedly shooting new branches off the old branches, and this is a little like the way web-links (and mental associations) form. Viewed as a geometrical structure, this kind of tree is hard to draw, for if you try and draw a densely branching tree on a piece of paper, you quickly run out of room. Some of the lines end up crossing over the other lines. (To fit the extra lines in we can either ask for more room or we can bump our drawing up into higher dimensions.) But it's easy to imagine such a tree. And the Web lends itself to representing a highly branching mind-tree because we can stick in as many links as we want.

But before committing to the idea of a tree, let's think a bit about clouds. When I say that a cloud is a fractal, I have in mind a model in which we think of a cloud as a certain shaded volume of space. This shaded cloud region has a very complicated shape, with lots of holes and tendrils. One way to imagine mathematically constructing a cloud is to start with a cube of space and to then subdivide it into, say, twenty-seven subcubes (cutting it in three along each dimension, like a Rubik's cube.). Remove each subcube that doesn't have any of the cloud in it. Then take each of the remaining subcubes and divide it into twenty-seven subsubcubes. Again remove the pieces that don't touch the cloud. Repeat the process of dividing and winnowing out for a number of levels. If done in a regular fashion this can lead to a regular fractal such as the "Menger Sponge."

**[Show The Menger Sponge. (Image is in Rudy Rucker, Mind Tools (Houghton-Mifflin 1987)]**

Of course you don't have to build clouds up in such a regular way. You can use a more random process for removing subcubes, and then you end up with something more natural in appearance.

Maybe a mind is as much like a cloud as it is like a tree. You have some vague notion (like a cloud seen from a distance), and then when you examine it more closely it breaks into a number of denser regions. And these chunks in turn break into smaller chunks.

Like the mindscape, the Web has a fractal quality to it. One starts out headed for topic A, but when you get to the page for A, you notice a link to topic B, and you go look at B before reading A, but on page B, you find a tempting link C that you just have to read first, and so on.

In some sense you never can get started drawing a true fractal like the Koch curve, because you always have to put in another bump before the bump you want to get to. This is similar to the experience you have when you try to fully explain any aspect of your mindscape. And this is an experience you can also have when you surf the Web.

The attractive thing about the Web as a model of the mind is that its a kind of "paper" where you never have to "run off the edge" or "run out of dimensions." You can always add scrollbars or links to give yourself more room.

Certainly at this point in history, the Web doesn't match the branching-tree structure of a real human mind, but a Web-like structure *could be tuned* to be a tree like this.

Or, again, if we want to think of the mind as being like a cloud, we can also think of a web page as being like a cloud. It's a collection of concepts, and many of these can be hyperlinked to further web pages.

In other words, we can either think of a web page as branching like a tree or as having denser regions like a cloud.

Whenever I present these ideas to people I get a lot of objections. Here are a few of them, with my attempts an answers.

**Objection 1**. Just because the Web and the Mind are like fractals doesn't mean they're like each other. A Koch curve, a tree and a cloud are fractals, but they aren't the same.

**Answer 1**. The Web is endlessly tunable. I'm not saying that the Web *right now* is like the mind. I'm saying that it should be possible to use the Web to make a good representation of a mind.

**Objection 2**. What's so special about the Web? Couldn't you use a very fat book with a lot of footnotes to present a similar kind of branching hypertext?

**Answer 2.** Indeed you can make a printed model of a big hyperlinked Web site. You might, for instance, print out the text content and the images, and use footnotes for the hyperlinks. But it would be hard to maintain and cumbersome to read. This question suggests an interesting analogy. A good Web model of, say, Johnny X, would be something like *The Encyclopedia of Johnny X*, with lots of cross-references from article to article. How might Johnny X generate the content and the links for such a book? We'll discuss some science-fictional methods for this next month.

**Objection 3**. A Web site is static. The essence of your mind is that it is continually changing and reacting to things.

**Answer 3**. If a Web site were really to be like a mind it would have to have a certain self-animating quality. It should "browse itself" and let you watch or, better, it should let you input things into it and watch it react. If you had the content and the links for a mind-sized website in place, writing some driver software in Java wouldn't actually be that hard. Imagine, for instance, a background search engine that would keep popping up new associations to things on the screen.

**Objection 5.** The Web is all interconnected. So actually it is more like one mind than like a lot of minds.

**Answer 5.** You could indeed think of the Web as society's mind. And then the most frequently visited sites are the public mind's obsessions. But there will be individual pieces of the Web that correspond more to one individual's mind.

Carmack quote from *Masters of Doom*, search for it below.

## *Loose Ends Not Done*

### *Chap 1*

Discussion of search as an element of computation. Oppose it to constructivism. The whole reason computations are hard are because they allow endless search.

Gut and compress my *Galaxy* online column web-mind shtick for a few fast concepts and use them in section 1.7 about the web.

### *Chap 2*

Thought experiment: what if there were no indeterminacy. Would the world seem any different?

Chaos vs. Big Aha: Isn't our chaotic world in its own way as random as the branch of a multiverse? If so what then of the Big Aha, if not why not?

Multiverse Downer: Every plane crashes, every lottery ticket is a winner. Nothing matters. Yet, the shape of the world could be the attractor. Cf. ocean waves: they're not random in the wild anything goes sense, they're random in the sense of being at a random location on the attractor.

Stress point that there probably *aren't* any naturally occurring simple computations. Everything real is complex.

### *Chap 3*

Get a picture of the growth stages of a plant.

Get a picture of activator inhibitor growing the bones of a hand.

Quote Helen Fox Keller. "We don't need a computer simulation of a cell. We already have a cell."

Mention Deutsch's remarks on complexity of 3D protein folding.

For the Artificial Life reproduction section, find the quote from *Sirens of Titan* about the pond of robot parts.

### *Chap 4*

Use the word “lambent.” A highlighted set of ideas to indicate the lambent spotlight of consciousness.

An artist like Bosch is able to internalize nature’s morphogenetic flows and calculate naturalistic forms of their own.

Stephen “Hippie” Gaskin’s remark about looking at the masts of sailboats rocking in the SF harbor and then “integrating” them into one mind pattern, and feeling a rush.

Exercises: do the same for waving tree branches; in a stirred-up emotional state look outside at swaying tree branches and view their motions as analogous to your own mood vibrations.

### *Chap 6*

For section 6.1. *Why* does it even matter if we call the processes in physics, biology, psychology, and sociology computations? So what? What do you gain? Tedious and worrying to imagine all those enormous computations. Win 1: explains why events take the forms they do. Win 2: Natural undecidability.

I cast NUH in the form “For any formal system F and natural system P, there are statements about P undecidable by F.” Wolfram didn’t use the F. He instead said that given a notion of “negation” of a state, for many P there are states G which are undecidable in the sense that neither state G nor state ~G ever occurs. If we view it in this neutral manner, there’s no reason at all to be excited about this fact. I mean why *should* a computation be complete?

### *Research*

Link to Wolfram’s Open Problems.

## *Sidebar Quotes*

### *1.1 Wheeler: Information vs. Computation*

“Paper in white the floor of the room, and rule it off in one-foot squares. Down on one’s hands and knees, write in the first square a set of equations conceived as able to govern the physics of the universe. Think more overnight. Next day put a better set of equations into square two. Invite one’s most respected colleagues to contribute to other squares. At the end of these labors, one has worked oneself out into the door way. Stand up, look back on all those equations, some perhaps more hopeful than others, raise one's finger commandingly, and give the order ‘Fly!’ Not one of those equations will put on wings, take off, or fly. Yet the universe ‘flies.’” — Charles Misner, Kip Thorne, and John Wheeler, Gravitation (W. H. Freeman, 1970), p. 1208.

### *1.1.1 Turing: Arithmetic Computation*

"Computing is normally done by writing certain symbols on paper. We may suppose this paper is divided into squares like a child's arithmetic book. In elementary arithmetic the two-dimensional character of the paper is sometimes used. But such a use is always avoidable, and I think that it will be agreed that the two-dimensional character of paper is no essential of computation. I assume then that the computation is carried out on one-dimensional paper, *i.e.* on a tape divided into squares. I shall also suppose that the number of symbols which may be printed is finite. If we were to allow an infinity of symbols, then there would be symbols differing to an arbitrarily small extent. The effect of this restriction of the number of symbols is not very serious. It is always possible to use sequences of symbols in the place of single symbols. Thus an Arabic numeral such as 17 or 999999999999999 is normally treated as a single symbol. ...

"The behavior of the computer at any moment is determined by the symbols which he is observing. and his "state of mind" at that moment. We may suppose that there is a bound *B* to the number of symbols or squares which the computer can observe at one moment. If he wishes to observe more, he must use successive observations. We will also suppose that the number of states of mind which need be taken into account is finite. The reasons for this are of the same character as those which restrict the number of symbols. If we admitted an infinity of states of mind, some of them will be "arbitrarily close" and will be confused. Again, the restriction is not one which seriously affects computation, since the use of more complicated states of mind can be avoided by writing more symbols on the tape.

"Let us imagine the operations performed by the computer to be split up into "simple operations" which are so elementary that it is not easy to imagine them further divided. Every such operation consists of some change of the physical system consisting of the computer and his tape. We know the state of the system if we know the sequence of symbols on the tape, which of these are observed by the computer (possibly with a special order), and the state of mind of the computer. We may suppose that in a simple operation not more than one symbol is altered. Any other changes can be set up into simple changes of this kind. The situation in regard to the squares whose symbols may be altered in this way is the same as in regard to the observed squares. We may, therefore, without loss of generality, assume that the squares whose symbols are changed are always "observed" squares.

"Besides these changes of symbols, the simple operations must include changes of distribution of observed squares. The new observed squares must be immediately recognizable by the computer. I think it is reasonable to suppose that they can only be squares whose distance from the closest of the immediately previously observed squares does not exceed a certain fixed amount. Let us say that each of the new observed squares is within *L* squares of an immediately previously observed square. ..."

— Alan Turing, On Computable Numbers, with an application to the *Entscheidungsproblem*," Proc. Lond. Math. Soc. (2) 42 pp 230-265 (1936-7); (A link to this paper online can be found at www.turing.org.uk/sources/biblio.html).

### *1.1.2 von Neumann: Computers*

"Conceptually we have discussed . . . two different forms of memory: storage of numbers and storage of orders. If, however, the orders to the machine are reduced to a numerical code and if the machine can in some fashion distinguish a number from an order, the memory organ can be used to store both numbers and orders."

— Arthur Burks, Herman Goldstine, and John von Neumann, "Preliminary Discussion of the Logical Design of an Electronic Computing Instrument," reprinted in John von Neumann, Collected Works, Vol. V, p. 35, Macmillan Company, New York 1963.

### *1.1.4 Masters of Doom, Physics*

"...after so many years immersed in the science of graphics, he [John Carmack] had achieved an almost Zen-like understanding of his craft. In the shower, he would see a few bars of light on the wall and think, Hey, that's a diffuse specular reflection from the overhead lights reflected off the faucet. Rather than detaching him from the natural world, this viewpoint only made him appreciate it more deeply. 'These are things I find enchanting and miraculous,' he said, 'I don't have to be at the Grand Canyon to appreciate the way the world works. I can see that in reflections of light in my bathroom.'"

— David Kushner, *Masters of Doom*, (Random House, 2003) p. 295. Kushner is describing the programmer John Carmack, who developed most of the code for the first-person-shooter computer games Doom and Quake.

### *1.1.5 Hegel: Organic Growth --- or Society*

"For the rest it is not difficult to see that our epoch is a birth-time, and a period of transition. ... This gradual crumbling to pieces, which did not alter the general look and aspect of the whole, is interrupted by the sunrise, which, in a flash and at a single stroke, brings to view the form and structure of the new world.

"But this new world is perfectly realized just as little as the new-born child; and it is essential to bear this in mind. It comes on the stage to begin with in its immediacy, in its bare generality. A building is not finished when its foundation is laid; and just as little is the attainment of a general notion of a whole the whole itself. When we want to see an oak with all its vigour of trunk, its spreading branches, and mass of foliage, we are not satisfied to be shown an acorn instead. In the same way science, the crowning glory of a spiritual world, is not found complete in its initial stages. The beginning of the new spirit is the outcome of a widespread revolution in manifold forms of spiritual culture; it is the reward which comes after a checkered and devious course of development, and after much struggle and effort."

Georg Hegel, The Phenomenology of Mind (1807), Harper & Row, 1967, p. 75 - 76.

## *Outtakes*

### *Short Chapter-by-Chapter Outline*

(Note that this is an outdated chapter sequence that matches the initial proposal.)

1. Computers have evolved gradually, almost like a new animal species. By looking at how they do some common tasks, we discover a rigorously logical universe.
2. Our machines can simulate the workings of the nature so accurately that some scientists believe the world is fundamentally a set of computations.

3. Following Stephen Wolfram, if we look at randomly chosen computations, then we find there are only four basic types. In a certain sense, a fluttering leaf is as complex as a human mind.
4. If we peer into the mind, we discover a fractal structure that we can try and model by a Web-page-like program that I call a "lifebox".
5. The pixels of a computer screen can act as beautiful living skin. The virtual reality that people used to talk about has arrived as the hypnotic new art form of computer games.
6. There's a close analogy between computer programs and our own DNA. We can create evolving forms of artificial life within our computer worlds.
7. There's hope of one day evolving computers to listen and speak like humans. At this point we ourselves may become a new kind of species.
8. Thanks to computers we have the wonderfully anarchistic planetary telepathy of the world wide Web.
9. And the new science of quantum computation may help provide a deep scientific understanding of the human mind.

### *Outline of Chapter One*

(This was an outline of Chapter One as of February 9, 2004.)

In this initial chapter, we have the following sections. "1.1: Computation Everywhere" suggests that any rule-based process is a computation, "1.2: Unpredictability" explains how a computation, although rule-based, can produce surprise. Here we also introduce Wolfram's useful taxonomic notion of four classes of computation, "1.3:Reckoning a Sum" shows how to few human arithmetic calculations as computations; "1.4 Universal Machines" gives a brief history of electronic computers; "1.5: Turing Machines" introduces a useful early model of computers; "1.6: How a Computer Works" gives a quick overview of the innards of your desktop machine; "1.7: The Web" discuses the planetary computation being carried by our Internet; and "1.8: Cellular Automata" describes the best-loved paradigm for parallel computation.

### *Bitching About Computers*

Occasionally I see an old movie from the Thirties, or Forties or Fifties — and I'm always struck by the absence of computers. How peaceful things look without those demanding machines; how leisurely and free those old-time actors seem.

When they're done work for the day, they leave their office — without having to wait through any kind of computer log-off procedure. When they arrive at their offices in the morning they get on with their business — without spending an hour or more combing through email.

### *Quagmire*

And, don't worry, I won't short you on the fun of wallowing in any enticing intellectual quagmires that we do run across.

### *My First Computer*

The first computer I saw was when I was in high-school in Kentucky, maybe forty years ago. I was in the math club, and our group got a tour of the computing center at the University of Louisville.

The thing that impressed me most on that first computer excursion was a large mechanical drawing device --- they called it a plotter. The plotter was shaped like a sand box; it was a square hollow frame four feet by four feet. In the middle of the plotter was a bracket-mounted pen held in place by tight pulleys. The pen was supposed to draw on a three-foot square of paper, but at the time of our visit the plotter was out of order --- nothing unusual for computer hardware then or now.

Even though the plotter wasn't working, the idea of it stayed with me. By speeding up and slowing down two little pulley motors according to some equations, a computer could move a pen around to draw pictures. Some of the pictures were on the walls. I seem to remember a spider-web and perhaps a mechanical drafting of a nut or a bolt. The friendly white-shirted nerds who ran the computer center tried to explain how they used punch-cards to feed simple equations to the computer that ran the plotter — but their explanations didn't make any sense to me. The important thing was the visual evidence of the plotter: A computer could control a device capable of generating smooth shapes. Computers were more than punch-cards and teletype print-out.

In math class at that time — I think this would have been the eleventh grade —— we were studying equations with things like cube roots, exponentials and polar coordinates. I wished I had a plotter to play with, so as to quickly turn lots and lots of equations into pictures. I also wished that I could understand how to use punch-cards to put equations into a computer. I was hoping there might be a way to learn this without having to become a complete Martian like the computer-center guys.

### *Intro: Why I Hate Computers*

Computers are annoying. They require a huge amount of upkeep and then don’t really work the way you want them to; they create endless make-work tasks; they strain your body; they allow strangers to invade your privacy. They buzz. They’re ugly.

Working with computers isn’t quite like biting the head off a live chicken, but it’s close. The thing is, computers are somewhat repellent. The first thing that the first computer ENIAC did was to help design the hydrogen bomb. Computer cases are a dull, ugly shade of beige. Computers are the tools of telemarketers, dot-commers, oppressive governments, and digital snoops. Many of us have office jobs where using a computer is part of the daily grind. The damn things never work like you expect them to for more than a few weeks at a time. You have to constantly upgrade their software and hardware. Over-using a computer can damage your hands, your arms and your back. They flicker and they make an ugly noise. And so on.

### *Intro Bio: What I Use Computers For*

For many of us there’s a few “killer apps” that make our machines indispensable; these might include email, word processing, web browsing, games, database management, spreadsheet manipulation, graphic design, 3D drafting or, for a few, writing computer programs.

I often spend an hour in the morning on email, staying in touch with friends, family, students, and colleagues, also sometimes answering questions from the readers of my books.

As a popular science writer and science-fiction novelist, I'm quite dependent on my word processor. My web browser is my first resort whenever I have to research a topic; even though the web information is often too shallow to be fully satisfying, it at least points the way for further research. As a computer science professor, I'm constantly creating new programs for my classes. I also design web pages for my classes, and use spreadsheets to track my student's grades. Sometimes I scan in pictures and reformat them in a photo editing program. Once in awhile I use a computer algebra program to simplify or to graph some equations I'm interested in. And, now that I teach a course of computer game programming, I've begun playing more videogames, either on my computer or on a special-purpose console (really a computer as well) connected to my TV.

### *Hyperspace Topics*

Higher dimensions. Visualizing the hypercube. Curved Space. Life inside a Klein bottle.

### *Drafts of the Preface*

*Computation and Reality* discusses the relations between computers and several kinds of reality: the physical world, human society, and the life of the mind.

- World. Computers' ability to simulate nature have brought a new kind of science. It turns out to be very useful to think of the world as actually *being* a computation.
- Society. The daily news is unpredictable precisely because society is a parallel computation. The Web is our emerging planetary mind.
- Mind. It seems in principle possible to evolve a specific program to mimic almost any given human behavior. But introspection suggests that we have a "quantum mind" transcending any digital machine. Quantum computation may circumvent this.

The book is meant to be immediate and accessible rather than technical and comprehensive. Wherever possible, I base my examples on typical daily experiences, and on things you can observe happening in your own mind.

I'm also interested in a fourth level of reality: the *cosmic* one. *Computation and Reality* is intended to bring enlightenment as well as understanding.

Perhaps you're inclined to protest, "What could be less reality-enhancing, life-affirming, or mind-expanding than computation?"

Admittedly, computers can be annoying. They require a lot of upkeep; they don't do what you want; they strain your hands; they compromise your security. They buzz. They're beige. And so on.

Ah, but if you look, deep meanings float just beneath the pulsing screen. The computer is a bit like a microscope, a device for peering into new worlds. What makes it even more fun is that this particular seeing-tool can focus upon itself.

How did I come to write *Computation and Reality*?

As a teenager, I imagined that I'd become a philosopher, a physicist, or a beatnik writer. I ended up getting a Ph.D. in mathematical logic from Rutgers University.

My wife Sylvia and I raised a family, with me working a dual career as a math professor and an author. As well as thirteen science-fiction novels, I published three popular books about mathematics: *The Fourth Dimension*, *Infinity and the Mind*, and *Mind Tools*.

By the mid-1980s, I sensed something new in the air. I grabbed an opportunity to move to Silicon Valley, retooled, and become a computer scientist at San Jose State University. At the time, my writer friend Gregory Gibson said something prescient. "Imagine if William Blake had worked in a textile factory. What might he have written then?"

As a CS professor and a sometime software engineer for Autodesk Inc., I've spent nearly twenty years in our dark Satanic mills. I'm covered in a thick lint of bytes and computer code. And now I'm stepping into the light to share what I learned from the machines.

***

You might say that *The Lifebox* is a book about the philosophy of computer science.

"Lifebox" is a word I invented to describe a certain (not yet existing) computer-based invention. As a science-fiction writer I sometimes try out new ideas in the context of my stories and novels. But I don't think the actual lifebox will be very long in coming.

My idea for the lifebox is that it'll be a small interactive device to which you can tell your life story. It'll prompt you with questions and organize the information you give it. You can feed in as many digital images as you like as well.

Once you get enough information into your lifebox, it'll become something like a simulation of you. Your audience will be able to interact with the stories in the lifebox, interrupting and asking questions.

Why would you want to make a lifebox? Immortality, ubiquity, omnipotence. You might leave a lifebox behind so your grandchildren and great-grandchildren can know what you were like, or you might use your lifebox as a way to introduce yourself to large numbers of people right now, or you might let your lifebox take over some of the less interesting duties in your daily routine, such as answering routine phone calls and email.

My plan for *The Lifebox* is that in the process of talking about the lifebox device, I'll manage to fit in discussions of most of my favorite topics in computer science.

Designing the lifebox is complicated by the fact that a person's recollections aren't linear; they're a tangled tree of branches that split and merge. The mind, in other words, is a kind of fractal. Another way to put it is that the lifebox will be structured something like a web page, with links hooking your memories together.

Suppose that we look a bit further ahead, and try to imagine ways to "animate" a lifebox so that it begins putting together fresh ideas in your style. This will bring into play notions of chaos theory, artificial intelligence, evolutionary algorithms, and some of the notions randomness discussed in Stephen Wolfram's *A New Kind of Science*.

We might naturally wonder if a lifebox could ever become entirely equivalent to a human mind. One barrier is the common feeling that the mind as one immediately experiences it is quite unlike the working of a computer. But there is some reason to believe that the new concept of "quantum computation" could serve to provide a model of a physical system more like the mind as we know it "from the inside."

My goal in writing *The Lifebox* is to offer new concepts with which to understand the natural world, the human mind, and ultimate reality. And it wouldn't be too much to say that I'd like for *The Lifebox* to show non-technical people how to use ideas about computers as a tool for enlightenment.

Perhaps you're inclined to balk. Perhaps you're inclined to say, "What could less reality-enhancing, mind-manifesting, or philosophical than computers?"

### *Geek Joke Explained*

I was in fact tempted to call this tome *Early Geek Philosophy*. As many will know, in early 20th century American slang, "geek" had the specific meaning of a carnival sideshow performer who would bite the head off a live chicken or eat raw liver. In practice, a carnival geek was a person whose primary skill was a high tolerance for grossness, ridicule, and pain. By the late 20th century, with carnivals a rarity, a "geek" was simply a person who seemed very different from other people, usually a person lacking in social graces. Because so many of these types of people became involved with computers, "computer geek" became a natural name for programmers in general. Given that computers have been around for considerably less than a century, in the grand historical view of things the computer mavens of our time are still *early* geeks, so call me Rucrates!

Working with computers isn't quite like biting the head off a live chicken, but it's close. The thing is, computers are somewhat repellent. Computer cases are a dull, ugly shade of beige. Computers are the tools of telemarketers, dot-commers, oppressive governments, and digital snoops. Many of us have office jobs where using a computer is part of the daily grind. The damned things never work like you expect them to for more than a few weeks at a time. You have to constantly upgrade their software and hardware. They flicker and they make an ugly noise. A lot of us lost money on computer stocks in the Dot Com Bubble. And so on.

Who but a chicken-head-biting geek could stand to spend much time with such machines? What could less life-affirming, mind-manifesting, or philosophical than computers? Ah, but if you look, the secrets of life float just beneath the pulsing screen.

### *Self-Indulgent Waffling About My Opinions*

I'm expecting that after I finish writing *The Lifebox, the Seashell and the Soul* I'll know exactly what to think. I am, if you will, initiating an extensive two-year mental "computation" whose final results I can't predict. Of course, as you read this, you have the option of glancing at the preface or flipping ahead to the last chapter's conclusion — but these are shortcuts that, on October 8, 2003, I don't have access too.

### *False Start on Chapter One, May Have Usable Intro Stuff*

As a teenager, I imagined that I'd like to become a philosopher. My best friend and I agreed that what we'd most like to do would be to get college degrees in philosophy and spend the rest of our lives as bums talking about the meaning of life.

I've always had those two drives: the attraction to logical clarity, and the craving for ecstatic enlightenment. Faced with a purely intellectual experience, I'll wonder if the ideas can change the nature of my immediate experiences. And when I'm in the midst of daily life, I dream of ways to abstract the phenomena into scientific theories.

Science-fiction is of course a particularly congenial field for someone interested in mixing science and reality in a not-too-rigorous way — and I've published a dozen or more science fiction novels. But I do still have that love of exactitude. I have a Ph.D. in mathematics and I've published three somewhat philosophical books about mathematics: *Infinity and the Mind*, *The Fourth Dimension*, and *Mind Tools*.

Each of those earlier non-fiction books was devoted to a specific theme, and so is *The Lifebox, the Seashell, and the Soul*. The earlier books treated infinity, higher dimensions, and information. And the book you hold is about computation.

### *Levels of Software (The Tao of Duh)*

To be quite general, we might let the system software includes all the other software hidden in the machine. At the lowest level there's something called microcode that lives on the computer chip and tells it how to interpret machine-language instructions. A step up is the BIOS (Basic Input/Output Services) software, which also lives on a chip. The BIOS helps the computer wake up and begin reading the software that lives on the hard drive. At a third level of system software we find an operating system such as Windows, Linux, or the Mac OS.

Taken together, the various layers of system software are, if you will, like the unconscious, memorized knowledge of the machine. And the application software is like the specific instructions for the specific task at hand.

Ordinary users (it's interesting that society employs the same word for computer owners and drug addicts!) rarely change their system software. The only way to get new microcode is to physically install a new processor chip. Changing the BIOS involves a perilous process called "flashing the EPROM," and is normally attempted only by true geeks (we, of course, love doing it). And even changing the operating system, e.g. by succumbing to a Windows "upgrade," has a substantial probability of turning your computer into a doorstop.

***

Returning to the notion of hardware, in our personal computers we distinguish between the microprocessor chip and the memory. In any generalized kind of rule-obeying system, if we can single out some element that embodies the action of the rules, we'll call this element the *processor*. And, by the same token, if there's some element of the system that retains traces of the system's states, we'll call this the *memory*.

### *Bohr's Walking Stick, Illustrating Stored Program Concept*

In this connection I'd like to mention an analogy I once read in an essay by the physicist Niels Bohr. The essay was about the relationship between observers, their experimental apparatus, and the reality they investigate. But the pattern is the same.

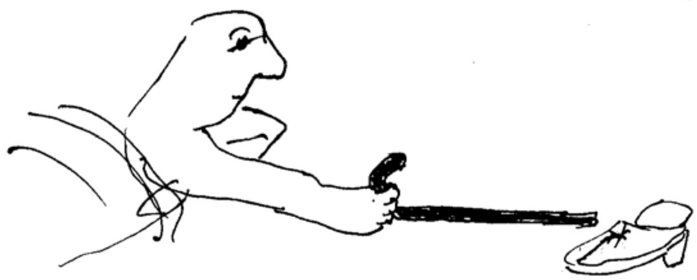


**Figure: Two Ways to Slice It.**

(Hiker + Stick) feeling (Shoe).
(Hiker ) feeling (Stick + Shoe).
(System Software + Application Software) acting on (Input).
(System Software) acting on (Application Software + Input)

Suppose I'm in a cold, pitch-dark room in the mountains and I want to arrange my shoes so I can step right into them as soon as I get out of bed. Without getting up, I feel around with my hand on the floor beside the bed. I don't feel the shoes, but my hand happens to light upon the handle of a light-weight walking stick. Good. Now I reach out into the darkness with the cane hoping to contact the shoes. On the left, my stick touches the leg of a chair, I trace around for the opening under the chair, poke in there, still no shoes. I reach over to the right, and come across something soft: my coat. And beyond that I find something firm but not stiff: one of my shoes.

As I'm feeling around with the stick, I almost forget that the stick isn't part of my body. It's as if my nerves grow out into the stick. I become sensitive even to the roughness and smoothness of the things my stick touches. Is the stick part of me, or part of the world? A little of both. The point is that the boundary between me and the non-me is somewhat fluid.

Application software is a "stick" that the system software wields in order to do things. Is the application part of the data or part of the computing device? A little of both.

### *Repeated Definition of Computation*

With all these considerations in mind, I can restate my definition of a computation one more time.

- *Definition*. A *computation* P is a process governed by a finitely describable set of rules called the *software*.
- The process takes place on a physical system called a *computer*, which can also be called the computation's *hardware*. We think of the physical computer's rules of behavior as a part of the software.

- The computer is thought of as embedded in linear time. The states of the computer can be called *inputs* or *outputs*, with earlier states being inputs relative to the later states, and the later states being outputs relative to the earlier states.
- The computational process that results from starting P on input In is called P(In), and if the process is in the output state Out at time t, we write *P(In, t) = Out*. The output state Out is completely determined by the software of P, the input In, and the amount of elapsed time t.
- In general, if P(In, t) = Out for any time t, we say *P(In) produces Out*. In other words, the computation P given input In is thought of as a process P(In) which produces a steady flow of outputs.
- If P(In) enters the state Out and stops changing, we say that P(In) *returns* Out. In this situation we also write P(In) = Out.
- Some computations P are accompanied by a *target detector* IsPDone which is a helper computation that allows us to distinguish certain *target states* as being states in which P has produced an answer. We require that for any state Out, the target detector IsPDone(Out) returns either True or False. In the case where P(In) produces Out and IsPDone(Out) returns True, we can say that *P(In) halts relative to IsPDone*.

### *Predictability*

Yes, if I multiply 318 times 478 with pencil and paper, the process is deterministic, but I don't know the answer till I carry out the calculation, so in that sense the answer wasn't predictable. Of course if you have a pocket calculator, you can figure out my answer faster than me. But, until you push those buttons, you don't know the answer either.

***

All four abstractly possible combinations of feasibility and predictability can arise, as suggested in the figure below. I made the picture messy to indicate that its not so easy to decide whether a computation is predictable or not. [*Author's note: Bull. I made it messy because I screwed the drawing up.*] (The meaning of the "Class" numbers will be explained in a few paragraphs.)

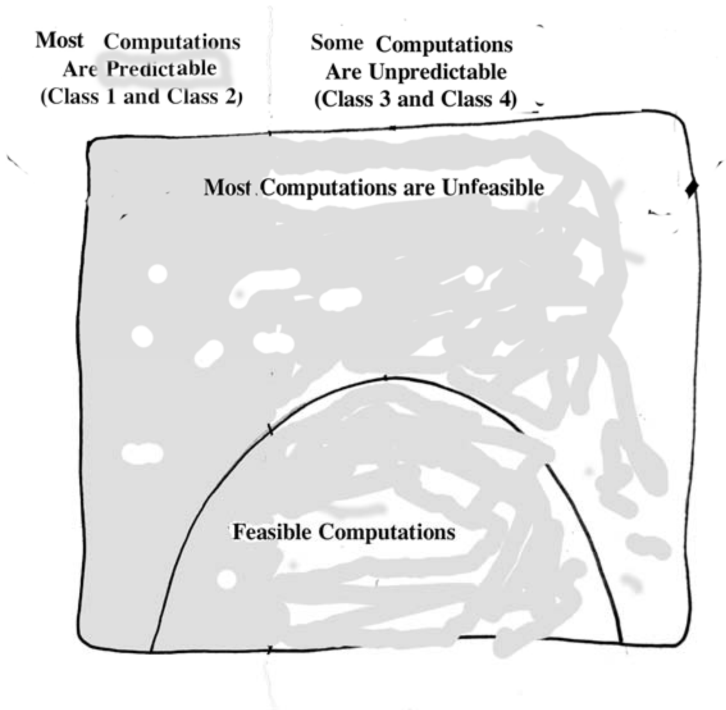



**Figure: Unpredictability and Unfeasibility.**

***Tractability***

Computer scientists use the word *tractable* as an approximate synonym to feasible. The usage difference is that "tractable" is more commonly applied to problems rather than to computations. That is, you might say a given problem is tractable if there is a feasible computation for solving it. And if a problem is *intractable*, this means that all the possible computations for solving it are unfeasible.

***Was Mrs. Graves my First Grade Teacher?***

Or was it Mrs. Devine? I remember one year being in class upstairs in Hilliard hall, the class where I blew the cow horn and got in trouble, and where he had to chew dry grass if we said, "Hey," and repeat "Hay is for horses, not for men." Don't remember that teacher's name. We were reading in there, though, a little bit. Then I had two years downstairs, one with Mrs. Graves, one with Mrs. Devine.

[I'm just going to combine the two.]

***Turing-style Details about Pencil and Paper Arithmetic***

- The paper is effectively divided into a grid of square cells, some containing symbols.
- The symbols are the digits 0 through 9, the decimal point, the arithmetic operation symbols +, -, ×, ÷, and % for percent.
- The inputs and outputs are patterns of symbols on the paper grid. Only certain,

orderly, "well-formed" patterns can actually serve as inputs. The reckoner can easily recognize these acceptable inputs, which are what we call arithmetic problems.

- Corresponding to each arithmetic problem is a standard procedure for the reckoner to follow, and the reckoner knows by heart which procedures apply to which kinds of problems.
- In applying an arithmetic procedure to an input, the reckoner begins a process which produces a series of successive patterns. This process is called a calculation. The reckoner can easily determine when the calculation is over, and the pattern on the paper at this time is the target output, that is, the answer to the arithmetic problem.

To complete these basic observations, here's a list of various kinds of elementary steps that are hooked together to make up the standard arithmetic procedures used to generate the reckoner's calculations.

- Reading symbols and noticing blank cells.
- Shifting attention from cell to cell.
- Remembering intermediate results.
- Using memorized look-up tables for single digit sums like 5 + 4 = 9.
- Writing symbols.
- Recognizing when the procedure is over.

### *Two Premature Turing Machine Explanations in Chap One*

I'm having trouble not putting this in. But I really don't need to, and it's hard, and there's no reason to mention it. Maybe later in Chapter 3, I can talk about it.

***

**Newer, simpler version.**

Starting with simple thoughts about arithmetic, Alan Turing formulated a definition of computation in the 1930s — well before any real electronic computers had been built. Turing's approach was to describe an idealized kind of computer called a Turing machine. A Turing machine is similar to a human reckoner, but without all the squishy stuff on the inside. A Turing machine has only some finite number of internal states. These are analogous to a reckoner's mental states, such as remembering to carry a one.

As a further simplification, a Turing machine uses a linear tape of cells instead of a two-dimensional grid of paper. A Turing machine focuses on one cell at a time on its tape. During each update, it reads the symbol in the cell, possibly changes the symbol in the cell, shifts its attention one cell to the left or one cell to the right, and enters a new internal state. Having completed one update step, it begins the next: reading the new cell, changing it, moving its head, and altering its internal state once again.

What determines the Turing machine's behavior? Look at it this way: each stimulus pair of <*internal state, read symbol*> leads to a unique response triple of <*write symbol, move direction, new state*>. We can think of the software for a Turing machine as being a lookup table that supplies a response triple for each possible stimulus pair.

The operating system for a Turing machine is the background machinery that causes it to behave in this way. To be more precise, we can say that a Turing machine's operating system forces it to cycle through the following three steps:

(Turing A) The machine reads the symbol that is in the active cell. It combines the read symbol with its current state to make a stimulus pair < *internal state, read symbol* >.

(Turing B) Given the stimulus pair < *internal state, read symbol* >, the machine looks in its program code to locate a corresponding response triple < *write symbol, move direction, new state* >.

(Turing C) On the basis of the response triple, the machine writes a symbol in the active cell, moves the active cell location one step to the left or to the right, and enters a new state. And then the machine returns to step (Turing A).

The software is the particular set of stimulus-response correspondences, while the operating system is the endless cycling through Turing steps A, B, and C. Furthermore, a Turing machine input is a string of symbols on the tape, and an output is any resulting pattern of symbols that appears on the tape at a later times.

The rudimentary quality of Turing machines makes it possible both to reason about their abstract capabilities, and to carry out exhaustive searches to find certain kinds of interesting Turing machines. The drawback is that not many interesting computations would be practically feasible for Turing machines, were we to actually build them. The Turing machine hardware is too simple and the expressiveness of the Turing machine software language is too limited. There's a good reason why Dell and Intel don't market Turing machines. Instead they market real computers.

***

**Older, clunkier version.**

In his 1936 paper, "On Computable Numbers," the British mathematician Alan Turing carried out a penetrating analysis of what makes a possible computation in the style of pencil-and-paper arithmetic. Recalling the old-style elementary school mathematics drill paper ruled into squares, he suggested that we can think of arithmetic as a process of reading, writing, and occasionally erasing and changing the symbols in an array of squares. Turing refers to the squares as cells — not in the sense of living cells, but in the sense of spreadsheet cells.

Next Turing suggested that we ought to be able to build a simple kind of machine to carry out this kind of computation — these are what we now call Turing machines.

Turing made his machines as simple as possible. He supposed that one of his Turing machines would have a read-write head which also remembers the machine's current internal state. He specified that each machine would have only some finite number N of states — the higher the N you use, the "smarter" the machine.

As a further simplification, Turing proposed that his machines could get by with a narrow tape of paper, a long row of cells. You could, for instance, have the tape act something like two-dimensional paper by using extra symbols [ and ] to mark the starts and ends of rows. Thus, at the beginning of computing the sum we mentioned above the tape might look like the following.

[ 2 7 5 ] [ 4 8 4 ] [ ]

If we set the adding machine's head down to the left of the first [, its course of action might go like the following, with the machine using internal states to remember which task it was doing and which digits it was moving around.

"Move right to the first ], go left one cell and read 5. Remember the 5 (by using an internal state) and move right to the second ], go left one cell and read 4. Add the remembered 5 to the 4 and get 9. Then move right to the second ], erase the ], write 9 in this spot, move one cell to the right and write ]. Now remember that you're working on the *second* column from the right. Move left to the first ], move right two digits (for "second column"), and read 7. Remember the 7 and move right to the second ], etc."

Eventually you'd could end up with the head back to the left of the first [, in an "I'm finished" state, with the tape looking like this.

[ 2 7 5 ] [ 4 8 4 ] [ 7 5 9 ]

So that's more or less how a one-dimensional tape would work.

We think of a Turing machine as having internal states 0, 1, 2, ... ,N for some finite N. We'll use state 1 as the starting state, and state 0 as the "halted" state. Once the machine enters state 0, it doesn't do anything else.

A Turing machine has an inexhaustibly long tape divided into cells. The head is always at some specific position on the tape, and we call this spot the active cell. To begin a computation, you mark some symbols into the tape cells, and set the machine's head to the left-most marked cell in state 1.

The Turing machine's computation consists of the following steps, repeated over and over

(TM-1) The machine reads the symbol that is in the active cell. It combines the read symbol with its current state to make an input *In* that consists of a pair <Current State, Read Symbol>.

(TM-2) Given the input pair *In* = <Current State, Read Symbol>, the machine looks in its program code to locate a corresponding output triple *Out* = <Write Symbol, Move Step, New State>.

(TM-3) On the basis of the output *Out*, the machine writes a symbol in the active cell, moves the active cell location one step to the left or to the right, and enters a new state between 1 and n. And then the machine returns to step (TM-1).

### *Footnote on Turing Machine States*

One additional point needs to be made regarding Turing machines and my general definition of a computation. In my definition, the computing system is viewed as an integral whole, with the outputs simply being states of the whole system. But Turing machines are described as having both an *internal state* and a "*tape state*" which consists of the current pattern of symbols on the machine's tape. But I can make Turing machines fit my general definition by viewing a Turing machine as having a *full state* which consists of an <internal state, tape state> pair. Suppose that a Turing machine M has a special start state S and a halt state H. If tapein is some tape pattern you want to compute on, you are really computing M(<S, tapein>). You are checking for the halt condition by applying an IsMHalted test such that IsMHalted(<internalstate, tapeout>) just checks if internalstate is H.

### *Beige Box Details*

The memory, often called RAM for "random access memory" can be imagined as a long ribbon of cells. The PC's memory cells hold so-called words of memory. Here "word" does not mean "meaningful language unit," it simply means a particular fixed number of bits, say, where a bit is of course a simple 0 or 1. In the 1980s a word was sixteen bits, then it became thirty-two bits, and in the 2000s it began phasing into sixty-four bits.

The memory addresses run from zero on through the thousands, millions and billions, depending on how much RAM the particular machine has. The "random access" aspect of the memory has to do with the fact that each memory cell has an address, and the processor is able to read or write the contents of a cell at any desired address without having to traverse all of the intermediate cells. If necessary, a computer can actually use more memory than it has in its RAM by reading and writing to extra disk, which function as a slower kind of RAM. (To read something off a disk, you need to physically move a read head to a certain address position. Reading from a given location on a RAM chip is more like pulsing a signal into an address wire and getting an answer back right away.)

***

How does the processor know how to carry out instructions? This is handled by a tiny program, known as *microcode*, that is coded right into the processor. Microcode interprets strings of bits as instructions to do things with the registers. The microcode lets the processor behave as if a certain sequence of bits means, say, "add the contents of register BX to the contents of register AX." The physical logic of the etched-in circuitry has tiny adding machines, multipliers, logic gates, and the like.

### *Confession of Anxiety in Chapter One*

Right now, writing this, I feel uneasy. I want *The Lifebox, the Seashell, and the Soul* to be a fun book, and here I am talking about arithmetic. And now, as if that's not boring enough, I've got to start in on tax forms? Jesus wept.

### *Rap about Halting Being Bad*

Your experience with pencil-and-paper computations, such as multiplying numbers, may have left you with the impression that a computation has to terminate with an answer. But this need not be true. A computation can be ongoing, with no specific termination time contemplated — think of a flowing river or of a computer screen-saver. Each of the successive states is an output linked to a moment in time.

Speaking of time, how long does a computation continue? We won't impose any bound at all. Certain kinds of computation will indeed signal when they've arrived at a desired result — for instance by beeping or by printing a result — and then halt in the sense of no longer changing their states. But there's a sense in which such computational processes continue after their halting point. It's just that after they halt they remain in a constant state.

For a computer that's supposed to calculate a number halting, is usually viewed as a good thing, but for a living being — which is also a kind of computation — halting has the bad connotation of death. Most naturally occurring computational processes are things that we like to keep going as long as possible.

### *Feasibility is Relative*

Feasibility is relative concept. To say a computation is feasible, you not only need to specify both the hardware and software of the computer system **M** that you plan to use as well as the amount of time **T** that you would be willing to wait for an answer. It would be more precise to speak of **M**-**T**-feasibility than simply of feasibility. Just to be precise, let's say that if I speak of feasibility, I'm normally thinking of **M** as being as good an electronic computer as currently exists on Earth, loaded with the best software algorithms we currently know of, and being allowed to run for a time **T** of one year.

### *Barlow and the Web*

The Web is a big deal. My friend John Perry Barlow, no slouch at self-promotion, used to give talks in which he'd compare the invention of the Web to the invention of fire. And then without flat-out saying so, John would manage to communicate an impression that he'd had something to do with the Web's creation.

### *Web Wakes Up*

What if we left all our machines connected and walked away, leaving them to communicate on their own? And what if each machine was itself running some kind of computationally rich program, making its own decisions about when to upload or download information. Might the Web as a whole wake up and start thinking?

### *Game of Life*

In order to compute a cell's new value, the Life rule first calculates what we can call the **EightSum** of the eight nearest neighbors other than the cell itself. And the cell's own value is looked at separately. The rule for Life can be specified follows:

- If the cell's value is 0 and its **EightSum** is 3, the cell's new state is 1.
- If the cell's value is 1 and its **EightSum** is 2 or 3, the new state is 1.
- In all other cases the cell's new state is 0.

### *Hodgepodge and RainZha*

The depicted instance of the Hodgepodge rule has a ready state 0, firing states 1 to 31, and a resting state 32. The update rule is as follows: (H1) If the cell is in the ready state 0 compute the sum S of its eight nearest neighbors. If S<5 then leave the cell in state 0, if 5<= S <100 set S to 2, and if S > 100, set S to 3. (H2) If the cell is in a firing state between 1 and 31, compute the average A of the cell and its eight neighbors, and set the cell to the value A + 5, rounding this down to 32 if A+5 > 32. (H3) If the cell is in the resting state 32, set it to the ready state 0.

The RainZha rule illustrated has the ready state 0, the firing state 1, and the 31 resting states 2 through 32. The RainZha rule is as follows: (R1) If a ready cell has either two or three firing neighbors it goes to the firing state. (R2) A firing cell goes to the resting state 2. (R3) A cell in a resting state between 2 and 31 goes to the next higher resting state. (R4) A cell in resting state 32 goes to the ready state 0.

### *Asynchronous Web*

A Web's scattered nodes sporadically send information back and forth over great distances. The Web's network architecture is a kind of parallelism, but without any kind of system-wide synchronization. Indeed, networks are often called *asynchronous*. The nodes are free to behave quite differently from each other, and each node has its own set of data, not all of which is necessarily shared with the other nodes.

### ***Summary of Chapter One***

Before moving on, I'd like to round off this chapter with a table to compare and contrast the traditional kinds of computation.

| | Human Reckoner | Turing Machine | Personal Computer | The Web | Cellular Automaton |
|---|---|---|---|---|---|
| Initial Input | Numbers, words and symbols on paper. | Symbols on the tape. | Files. | Names of networked machines. | Pattern of marked cells. |
| Interactive Input | Extra problems. | None. | Keyboard and mouse. | Requests to read and write data. | Reaching in to change some cell values while the computation runs. |
| High-Level Rule | A tax form or chain of math problems. May be read in. | A rule coded up as symbols on tape, can be read by a universal Turing machine. | Application software such as a word or image processor. May be read in. | Browsers, email, Web crawlers. | The rule used by the individual cell processors. |
| Medium-Level Rule | Person's learned reckoning behavior. | A lookup table matching stimulus pairs to response triples. | Operating system such as Windows or Linux | Applets, interactive Web pages, online data bases. | The run cycle which updates all the cells in synch. |
| Low-Level Rule | Algorithm for addition. | Standard run cycle: read, look-up response, write and move. | Microcode on the chip. The system clock | Communications software to send and receive data. | If implemented on a PC, the machine code that emulates the CA. |
| Lowest-Level Rule | Memorized plus and times tables. | Devices for reading, writing, and moving the head. | Chip architecture such as logic gates and adders. | The software on the networked machines. | If implemented on a PC, microcode and chip architecture. |
| Helper Rules | Ability to read and write. | How to read and writing symbols | The BIOS code for reading and writing files. | Read and write methods. | Graphics to display the CA. |
| Underlying disciplines | Psychology, biology, physics, logic. | Physics, logic. | Electrical engineering, physics, logic. | Communication theory, computer technology. | Depends on the implementation. |

**Table 1.3: PCs, reckoners, Turing machines, and the Web. Finally, to summarize this chapter, here's some of the key points I made.**

- A computation usually has *multiple levels* of rules, not all of them explicit.
- It's possible for two distinct computational processes to have *equivalent* behavior, but it may be that one is *faster* than another.
- Although a computation may be theoretically possible to carry out, it can be practically *unfeasible* to do so.
- The flow of many computations is in some sense *unpredictable*.
- A legitimate computation may run *endlessly* without having to reach a conclusion.

### *Joke about Von Karman Vortex Street*

As a sometime fiction writer, I like to think of Von Karman Vortex Street as a place, possibly located somewhere near Miles Davis's Green Dolphin Street and the Firesign Theater's Non-Euclid Avenue.

### *Fredkin's Billiard Ball Computer*

In the 1970s, the computer scientist Edward Fredkin gave a theoretical proof that, if you idealize away all of the real-world crud, you can make a universal computer from billiard balls bouncing around on a sufficiently large table. At the time this seemed like a surprising result, but by now it's beginning to seem likely that almost all of the physical systems we encounter are universal, just as they are.

Now, in reality, you can't actually build a Fredkin-style billiard ball computer, for the balls' large-scale motions will quickly display the slight inaccuracies in your starting conditions, not to mention the influences of effects as small as gravitational forces from your body as you walk around the table observing the experiment. The system won't behave in the repeatable digital fashion that you'd planned. It's not realistic to expect the motions of ordinary objects to behave like a digital computer. We live in an irredeemably analog world.

But Fredkin's proof-in-principle encourages us to believe some form of Wolfram's Principle of Computational Equivalence, which claims that essentially all physical systems embody universal computations fully rich enough to emulate anything that happens inside an electronic machine.

### *Schrödinger's Wave Equation*

I'm not going to try and explain Schrödinger's Wave Equation in detail, but I do feel I owe you a few comments.

- Since a system's wave function **ψ(x)** changes with time, we write it as **ψ(x, t)**.
- We put an arrow over the **x** to indicate that this symbol stands in for an arbitrarily long list of state variables
- Multiplying by **i** on the left corresponds to a ninety-degree rotation of phase.
- **h** is a the tiny number called Planck's constant.
- The quotient with the two backwards sixes means "the change with respect to time".
- The ***H*** on the right is the so-called Hamiltonian operator, which corresponds to the total energy of the system.

In practice it's difficult or even impossible to write out a precise formula for a complicated system's Hamiltonian operator — which fact tends to limit the real-world usefulness of Schrödinger's Wave Equation.

### *Randomness*

When you measure a system, its wave function undergoes an abrupt change — sometimes called the collapse of the wave function. The resulting simplified wave function is called an *eigenstate* of the measurement. The particular measurement eigenstate is picked wholly at random, although in accordance with probabilities that can be calculated from the pre-collapse wave function.

Regarding the randomness of quantum mechanics, where would the randomness come in? One might say that the world contains two kinds of computation: the computation of quantum mechanical physics and the computation of classical physics. Each of these is deterministic, but there is an unavoidable element of randomness in switching from the quantum mechanical view to the classical.

### *Dumping on Entanglement*

Entanglement is a popular topic for quantum mechanical mystery-mongering. But it's easy to read too much into it. If you heat a large flat pan of water, steam bubbles will appear at the same time on opposite sides of the pan. This doesn't necessarily mean that there's a magical faster-than-light entanglement between distant pairs of bubbles. It can simply mean that the bubbles are the result of a lower-level common cause.

### *Unfinished Summary of Chapter Two*

- Physical computations involve very many states, and we speak of them as analog. Although mathematics speaks of infinite ranges of real numbers, its enough to think of analog computations as involving very many possible values.
- A physical computation is usually *unrepeatable*.
- A computation can incorporate *quantum indeterminacy* and still have a definite rule.

### *Human language and the hierarchy of programming languages.*

By way of making automatism seem reasonable, I show how programmers have worked out a hierarchy of languages in order to bridge the gap between bytes and reality, and I explain some of the ways in which the real world can be simulated by software.

There are layers upon layers of languages used for controlling computers. The lower-level languages are more closely related to the machine, while the higher-level languages are more abstract, more appropriate for describing human thoughts.

Human language itself includes low-level and high-level features. Let's clarify the distinction by saying a bit about human language before we talk about the languages of the machines.

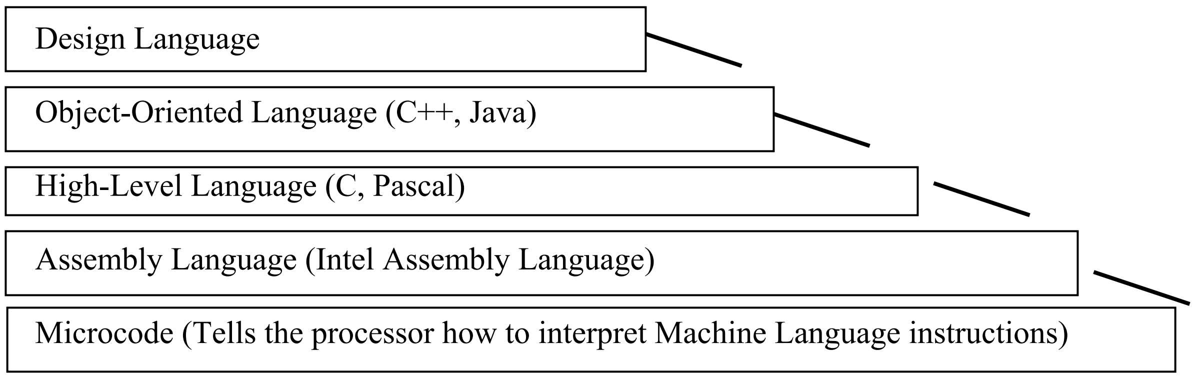
One of the first things a child learns is names for simple body sensations. Cold, wet, hungry, tired. And the names of objects. Mommy, hand, water, bed, sun. And simple verbs. See, stand, walk, cry. Slowly the child learns to make a few simple sentences.

Occasionally I've lived in a foreign country and made conversation in my hosts' language. When doing this, I often think a game of Lotto that my children had. The game consisted of a hundred pairs of glossy square cards, each card blank on one side and with a color photo of some object on the other side. Mommy, hand, water, bed, sun. Clumsily talking a foreign language is like having a pocketful of the Lotto squares and handing them to someone one by one. Tomorrow sun swim eat sausage.

By contrast, when you gracefully speak your native language, it's like singing or dancing. The ideas just flow out. And if you happen to be speaking with a close friend or a loved one, the language even begins to feel like telepathy. Your thoughts jump back and forth. Describing a conversation between two lovers in one of his books, Vladimir Nabokov writes something like, "We had one of those conversations in which the words leave no trace of memory. It was like a duet in an opera, the emotions riding on wings of song."

Before starting in on programming languages, I need to mention a distinction between the languages of humans and computers. In any given human society, children, poets, and lovers all use the same language; the difference is that they use smaller or larger parts of the language, with lesser or greater facility. But in the computer world there are strictly distinct layers of language.

We'll discuss five levels of computer language: microcode, assembly language, high level language, object-oriented language, and design language. The higher levels depend upon the lower levels.




**Levels of Computer Language**

### *Microcode and BIOS*

The microprocessor chip inside a desktop computer behaves a little like a moderately intelligent cockroach that scuttles about upon the long ladders of memory inside your machine. The cockroach microprocessor reads an instruction here, takes in a snippet of data there, massages the data according to the instruction, and then writes the altered data somewhere else.

The language that directs these inner workings of the microprocessor is called *microcode*. Microcode is the purview of heavy-duty computer engineers who work at the chip factories; generally a chip's microcode is developed in tandem with the chip's physical design.

If you think of the chip as a little city with electrons racing up and down its streets, then the microcode is like the program that controls this tiny city's traffic lights. The microcode is permanently etched into the silicon at the chip factory; it isn't something the ordinary user or even software programmer normally deals with or even thinks about very much.

There is one exciting (for geeks) exception. If you get deeply involved in upgrading and tweaking your computer, and start checking for hardware driver updates on the web, you may from time to time be able to download new "BIOS" code for one of the chips in your machine, perhaps for the main microprocessor or for a subsidiary processor that handles a task like graphics. And then, oh joy!, you can "flash the BIOS" which means copying new microcode onto the processor.

Flashing the BIOS is a slightly risky activity, perhaps on a par with open heart surgery. A worst-case scenario arises if your computer loses power while in the midst of a BIOS flash, like if you (duh!) turn off your computer or if you (whoops!) kick the plug out of the wall, or if (oh no!) there's a sudden power outage. In this event you might end up with half the old microcode and half the new microcode, and the two will no doubt be incompatible and your machine won't run at all. At this point, you'd call on the services of an ueber-geek to reflash the BIOS from scratch.

The instructions that the chip reads out of the memory are called machine language.

### *Everything is a Program*

Philosophical ontology and computer science's Object Oriented Programming (OO) as in C++ or Java.

Everything is mathematics → Everything is a set → Everything is an object.

Objects are better than sets, as they have function pointers, they are a verb as well as a noun.

An interesting thing about OO (object-oriented) languages is how they are like set theory. We build up class definitions from the primitive byte variables (of various standard lengths, recall that a floating point real is just four(?) bytes, for instance) like the way set theorists build up all of mathematics from commas and pairs of brackets. Objects are even better than sets, though, as they contain function pointers and corresponding blocks of code, that's a cool thing about the von Neumann architecture is that we have instruction pointer and data pointer, so we crawls over both data and code.

Software engineering. Teaches us to look for patterns in our lives.

### *Activator-Inhibitor CA Rule*

My rules have a definition of the following form, where I write a and b for activator and inhibitor levels.

(Avoid division by 0) IF (b > bMin) THEN bSafe = b ELSE bSafe = bMin.
(Activator) aNew = a + aDiffuse • (aNabeAvg - a)
+ sDensity•a•a/(bSafe • (1 + aSaturation•a•a))
+ sDensity • aBase
- aDecay • a.
(Inhibitor) bNew = b + bDiffuse • (bNabeAvg - b)
+ sDensity•a•a
- bDecay • b.
(Clamp) Clamp both a and b to be in the range [0, abMax].

### *Kaneko-Abraham Rule*

Their idea was to combine diffusion and the logistic maps as follows. Suppose I write pop for a cell's current population value, and newpop for the new value that I want to compute. At each update each cell computes a diffusion term and a poplogistic term based on a logistic parameter A and a diffusion parameter B. The parameters A and B are the same for each cell.

**diffusion = B•(Average of neighboring population values - pop)**
**poplogistic = Logistic(A, pop)**
**newpop = poplogistic + diffusion**

This approach produces a phenomenon called numerical instability, meaning that all the cell values have a strong oscillation from generation to generation, producing artifacts such as a persistent domain boundaries and a checkerboard pattern. Be that as it may, the rule is computationally interesting. The unstable generational oscillations are more or less in synch across all the cells, so patterns are in fact able to emerge. The figure in the text shows scrolls in a version of this rule where each cell looks only at its four nearest neighbors, we use a logistic growth factor A of 2.1 and a diffusion rate B of 0.8. (Note that our A value of 2.1 is what Ralph Abraham's paper would call 0.5025, as he multiplies his A by a factor of 4.)

In the figure shown in the text, the checkerboard pattern has been reduced both by showing the differences between cells and their neighbors and by coloring nearby values the same. The domain boundaries are visible as looping lines. Over time these loops will shrink and disappear, leaving one or more loops that wrap all the way "around" the simulation space which, as usual, identifies the left and right edges and the top and bottom edges.

To me, the Kaneko-Abraham method seems unnatural in that the diffusion and logistic operations are in some sense out of synch. Essentially each cell does a logistic update and then does diffusion based on its values before the logistic update. I would have expected better results from a rule in which all the cells diffuse their values first and only then do a logistic update.

**diffusion = B•(Average of neighboring population values - pop)**
**newpop = Logistic(A, pop + diffusion)**

This approach does give smoother, somewhat less unstable patterns, but I've been unable to find Zhabotinsky scrolls with it — at best I've come up with some Turing spots. As I discuss in the text, when I use this second approach to the algorithm with two species, I had good success.

### *Artificial Life*

Vaucasson's duck. The little writing manikins in the Neufchatel museum.

Alife lacks morphogenesis. The shapes really don't arise naturally from the code.

A life lacks homeostasis.

Improve DNA-like genomes by fitness proportional reproduction. Fitness = gnarliness?

The notion of evolution has been strongly used by computer scientists in attempts to find good programs. Fitness function.

### *Telerobotics*

(Lifted from my *Artificial Life Lab* manual.)

For many applications, the user might not need for a robot to be fully autonomous. Something like remotely operated hand that you use to handle dangerous materials is like a robot, in that it is a complicated machine which imitates human motions. But a remote hand does not necessarily need to have much of an internal brain, particularly if all it has to do is to copy the motions of your real hand. A device like a remote robot hand is called a *telerobot*.

Radioactive waste is sometimes cleaned up by using telerobots that have video cameras and two robotic arms. The operator of such a telerobot sees what it sees on a video screen, and moves his or her hands within a mechanical harness that send signals to the hands of the telerobot.

I have a feeling that, in the coming decades, telerobotics is going to be a much more important field than pure robotics. People want *amplifications* of themselves more than they want *servants*. A telerobot projects an individual's power. Telerobots would be useful for exploration, travel, and sheer voyeurism, and could become a sought-after high-end consumer product

But even if telerobots are more commercially important than self-guiding robots, there is still a need for self-guiding robots. Why? Because when you're using a telerobot, you don't want to have to watch the machine every second so that the machine doesn't do something like get run over by a car, nor do you want to worry about the very fine motions of the machine. You want, for instance, to be able to say "walk towards that object" without having to put your legs into a harness and emulate mechanical walking motions---and this means that, just like a true robot, the telerobot will have to know how to move around pretty much on its own.

There is an interesting relationship between a-life, virtual reality, robotics, and telerobotics. These four areas fit neatly into the Table 1-3, which is based on two distinctions: firstly, is the device being run by a computer program or by a human mind; and, secondly, is the device a physical machine or a simulated machine?

| | Mind | Body |
|---|---|---|
| Artificial Life | Computer | Simulated |
| Virtual Reality | Human | Simulated |
| Robotics | Computer | Physical |
| Telerobotics | Human | Physical |

Table 1-3: Four Kinds of Computer Science

Artificial life deals with creatures whose brains are computer programs, and these creatures have simulated bodies that interact in a computer-simulated world. In virtual reality, the world and the bodies are still computer-simulated, but at least some of the creatures in the world are now being directly controlled by human users. In robotics, we deal with real physical machines in the real world that are run by computer programs, while in telerobotics we are looking at real physical machines that are run by human minds. Come to think of it, a human's ordinary life in his or her body could be thought of as an example of telerobotics: a human mind is running a physical body!

### *Code Growth Analogy*

There is a sense in which computer programs are indeed grown: the executable machine code arises from compiling and assembling the original source code. Here the source code of the program is like the genome, the development environment is like the womb, and the resulting program is like the body. But really we want something more.

### *Walker's Brag About His Server's Homeostasis*

Here's a quote from an email from my ultrageek pal John Walker, where he's bragging about the stability of his email server.

> As you go upmarket from PCs, you find more and more capabilities which provide homeostasis-like behavior. The 1998 vintage Sun mid-range server which runs my Web site has 4 CPUs and 5 Gb of RAM, and has been running continuously for...let's see:
>
> vitesse:/files/kelvin> uptime
>
> up 209 day(s), 23:31
>
> All of the RAM has double error correction and the operating system emails me when a chip is getting too many correctable errors so I can replace it before it dies entirely (and even that doesn't crash the system).
>
> All the hard drives are mirrored onto two or three separate drives on two independent controllers, and any failure simply causes a hot spare to be pressed into service, and again an email sent to me to replace the failed unit. The internal temperature is monitored, and fan speed is adjusted to keep it within limits. If the ambient air is too hot, the CPU clocks slow down to keep the temperature within limit, and if that's not sufficient, the system will put CPUs into hibernation, falling all the way back to one if necessary. If the system does crash, it performs a hardware test at reboot time and can "boot around" any failed components as long as it has one working CPU, memory bank, and system drive. There are three power supplies, of which only two are needed to run. The next generation after my server has three separate power cords, which can be wired to separate circuits and UPS boxes. And all of these components can be changed without powering down the machine or halting the system. And this is just one step up from a PC, and slower than a 2 GHz machine of today.

This machine is of trivial complexity compared to a prokaryotic bacterium, not to speak of a metazoan organism, but a glimmering of homeostasis is there.

### *RNA Evolution*

As well as trying to understand the chemical reactions that take place in living things, biochemists have investigated ways of creating the chemicals used by life. It is now possible to design artificial strands of RNA which are capable of self-replicating themselves when placed into a solution of amino acids; and one can even set a kind of RNA evolution into motion. In one recent experiment, a solution was filled with a random assortment of self-replicating RNA along with amino acids for the RNA to build with. Some of the molecules tended to stick to the sides of the beaker. The solution was then poured out, with the molecules that stuck to the sides of the vessel being retained. A fresh food-supply of amino acids was added and the cycle was repeated numerous times. The evolutionary result? RNA that adheres very firmly to the sides of the beaker.

### *Scripting computer games.*

Virtual world and agents bouncing off each other.

The history of virtual reality, a fake world where things can interact asynchronously.. Three-dimensional graphics, immersion, and user interaction. Walking around a city talking on a cell phone.

The creatures in a computer game are an excellent simulation of parallel agents. The problem of creating a satisfying computer game is, in small, an image of the question of why it is we happen to live in such an interesting world. Thinking about game design gives us fresh insights into both metaphysics and to non-traditional approaches to story-telling.

Some may remember the excitement over "virtual reality" in the 1990s. Here in the 2000s, we have virtual reality all but fully implemented in our games. Games bring together the full panoply of computer technology including: simulations of physics, computer graphics algorithms, artificial intelligence, computer aided design, user interfaces, and web-based connectivity. In addition, games draw in such traditionally artistic endeavors as the visual arts, sound design, and story-telling.

The essence of a good gaming experience is forgetting about the outer world. Absorption. Being in the zone. The flow. But walking to the bakery can be as interesting as a computer game.

Possible games of the future.

### *Don't Worry About Biotech.*

I consider how serious are the threats posed by biotechnology, and what might be done about the problems. The news is mostly good. Our experiences with the virtual ecologies of cyberspace are numerically illuminating. I also explore the logic behind the open source genomics movement, which argues that it's safer to have many different people tweaking genes than to have this power in the hands of a few large corporations.

### *Class Four Time Series*

What would a class 4 time series look like? It's not so visual, as we're hopping around on a line. But we could have class 4 behavior if it condenses, say on a Cantor set, which would a paradigmatic example of a strange attractor embedded in a line. I think James Crutchfield was interested in looking for Class 4 behavior in time series generated by the logistic map.

### *Weinberg's Put-Down*

There's a good reason why Dell and Intel don't market Turing machines. Universality isn't everything.

### *Frankenstein*

The most famous fictional character who tries to create life is Victor Frankenstein, the protagonist of Mary Shelley's 1818 novel, *Frankenstein or, The Modern Prometheus*.

Most of us know about Frankenstein from the movie versions of the story. In the movie version, Dr. Frankenstein creates a living man by sewing together parts of dead bodies and galvanizing the result with electricity from a thunderstorm. The original version is different.

In Mary Shelley's novel, Victor Frankenstein is a student with a deep interest in chemistry. He becomes curious about what causes life, and he pursues this question by closely examining how things die and decay — the idea being that if you can understand how life leaves matter, you can understand how to put it back in. Victor spends days and nights in "vaults and charnel-houses," until finally he believes he has learned how to bring dead flesh back to life. He sets to work building the Frankenstein monster and finally reaches his goal:

> "It was on a dreary night of November, that I beheld the accomplishment of my toils. With an anxiety that almost amounted to agony, I collected the instruments of life around me, that I might infuse a spark of being into the lifeless thing that lay at my feet. It was already one in the morning; the rain pattered dismally against the panes, and my candle was nearly burnt out, when, by the glimmer of the half-extinguished light, I saw the dull yellow eye of the creature open; it breathed hard, and a convulsive motion agitated its limbs... The beauty of the dream vanished, and breathless horror and disgust filled my heart."

What writer hasn't experienced something like these feelings upon seeing the work which he or she ends up writing — as opposed to the original dream!

### *Architectures*

I find it useful to characterize a computing system's architecture in terms of the following two distinctions.

- How many simultaneous processes are taking place, *one* or *many*?
- How many different sets of memory are used, *one* or *many*?

And in the case where there is one shared memory set, we can make a third distinction.

- If there is only one memory set, is the data access of the individual process or processes *local* or *global*?

Some of the architectures that we'll be discussing are summarized in the table below. All of the cases with many processes and many memory sets seem to share the same network architecture as the Web, so I'll just write "network" to describe their data access.

| Example | Number of Processes | Number of Memory Sets | Access To Memory |
|---|---|---|---|
| Turing Machine | One | One | Local |
| Personal Computer | One | One | Global |
| Classical Physics | Many | One | Local |
| The Web | Many | Many | Network |
| Quantum Mechanics | Many | Many | Network |

| | | | |
|---|---|---|---|
| Organism | Many | Many | Network |
| Human Society | Many | Many | Network |
| The Mind | Many | One | Global |

Table 1.4: Some computer architectures

### *Von Neumann Self-Reproduction*

In the late 1940s, von Neumann gave some ground-breaking lectures on the topic of whether or not it would ever be possible for a machine, or "automaton," to reproduce itself.

Usually a machine makes something much simpler than itself — consider a huge milling machine turning out bolts. Could a machine possibly fabricate machines as complicated as itself? Or is there some extra-mechanical magic to self-reproduction? To simplify the problem, von Neumann suggested that we suppose that our robots or automata are made up of a small number of standardized parts:

> I will introduce as elementary units neurons, a "muscle," entities which make and cut fixed contacts, and entities which supply energy, all defined with about that degree of superficiality with which [the theory of neural networks] describes an actual neuron. If you describe muscles, connective tissues, "disconnecting tissues," and means of providing metabolic energy . . . you probably wind up with something like ten or twelve or fifteen elementary parts.[1]

Using the idea of machines made up of multiple copies of a small number of standardized elements, von Neumann posed his question about robot self-reproduction as follows.

> Can one build an aggregate out of such elements in such a manner that if it is put into a reservoir, in which there float all these elements in large numbers, it will then begin to construct other aggregates, each of which will at the end turn out to be another automaton exactly like the original one?[2]

Using techniques of mathematical logic, von Neumann was then able to deduce that such self-reproduction should in fact be possible. His proof hinged on the idea that an automaton could have a blueprint for building itself, and that in self-reproduction, two steps would be necessary: (1) to make an exact copy of the blueprint, and (2) to use the blueprint as instructions for making a copy of the automaton. The role of the blueprint is entirely analogous to the way DNA is used in biological self-reproduction, for here the DNA is both copied and used as instructions for building new proteins.

The complexity of a reservoir full of floating machine parts hindered von Neumann from making his proof convincing. The next step came from Stanislaw Ulam, who was

---

[1] From "Theory and Organization of Complicated Automata," 1949, reprinted in John von Neumann, *Theory of Self-Reproducing Automata*, University of Illinois Press, Urbana 1966, p. 77.

[2] From "The General and Logical Theory of Automata," 1948, reprinted in: John von Neumann, *Collected Works, Vol. 5*, Macmillan, New York 1963, p. 315. The weird scenario described in this quote is reminiscent of a scene in Kurt Vonnegut, Jr.'s *Sirens of Titan* where an unhappy robot tears himself apart and floats the pieces in a lake.

working with von Neumann at Los Alamos during those years. Ulam's suggestion was that instead of talking about machine parts in a reservoir, von Neumann should think in terms of an idealized space of cells that could hold finite state-numbers representing different sorts of parts.

Ulam's first published reference to this idea reads as follows:

> An interesting field of application for models consisting of an infinite number of interacting elements may exist in the recent theories of automata. A general model, considered by von Neumann and the author, would be of the following sort:
>
> Given is an infinite lattice or graph of points, each with a finite number of connections to certain of its "neighbors." Each point is capable of a finite number of "states." The states of neighbors at time $T_n$ induce, in a specified manner, the state of the point at time $T_{n+1}$.
>
> One aim of the theory is to establish the existence of subsystems which are able to multiply, i.e., create in time other systems identical ("congruent") to themselves.[3]

By 1952, von Neumann had completed a description of such a self-reproducing "cellular automaton" which uses 29 states. Von Neumann's CA work was not published during his lifetime; it seems that once he saw the solution, he became distracted and moved on to other things.

### *Wolfram On Discovering CAs*

It really is quite remarkable that, starting from an initial condition of a single marked cell, something as simple as a "Rule 30" can generate Class 3 randomness and that a "Rule 110" can generate a Class 4 process resembling a complex computation. In his own somewhat hyperbolic style, Wolfram expresses his excitement as follows.

> And what I found — to my great surprise — was that despite the simplicity of their rules, the behavior of programs was often far from simple. Indeed, even some of the very simplest programs that I looked at had behavior that was as complex as anything I'd ever seen ... I have come to view [this result] as one of the more important single discoveries in the whole history of theoretical science.[4]

Mathematicians have never discussed very much the fact that you can get complicated things from simple causes like CA Rule 30 or CA Rule 110. The one area where they have talked a bit about complexity emerging from simple rules is in the distribution of the primes and the irregularity of the digits of pi.

Wolfram expresses surprise over this in these words. "Often it seemed in retrospect

---

[3] From "Random Processes and Transformations," 1950, reprinted in : Stanislaw Ulam, *Sets, Numbers and Universes*, MIT Press, Cambridge 1974, p. 336.

[4] Stephen Wolfram, *A New Kind of Science*, p. 2.

almost bizarre that the conclusions I ended up reaching had never been reached before."[5] And then he gives some thought to why complexity has so long been ignored.

One reason seems to be that in normal technology we figure out what we want a system to do, and then reverse engineer it, and to be sure it will work, we stick to very simple and predictable systems. But nature is under no such constraint. It's in fact common for natural systems to achieve their effects by the unpredictable accumulation and interaction of a myriad of small computational steps. A second reason why many of Wolfram's ideas are historically new is that it's only the coming of computers that made his investigations possible. Historically, people simply never in history looked at things like Rule 30. The most complex patterns one tends to see in earlier art and architecture are fairly regular fractals three or four levels deep, such as, for instance, the curlicues of Celtic illuminations or iron-work vines.

### *Parallel*

But in point of fact, word-processing, emailing, and Web browsing all work just fine on single-processor machines. Electronic parallel computers *are* being built, but only for very intense supercomputing tasks like weather prediction or, depressingly, for nuclear weapons simulation.

***

And if there's more than one process, are they all *copies* of the same process, or are they *different* from each other?

Programming parallel machines is in fact much easier if you synchronize the processors and only let them write to nearby memory. If you let the processes run at their own speeds and read and write wherever they like, the situation can get exceedingly hairy.

***

I saw my first parallel hardware machine like this in 1984 at a Naval Research Center in Beltsville, Maryland. I'd been invited there to give a talk on the fourth dimension — maybe the scientists had some faint hope I might have an idea for a faster-than-light hyperdrive! At the time, I was just getting interested in cellular automata such as the game of Life; cellular automata are a particularly simple kind of parallel computation. In fact I was on my way back from interviewing Wolfram and the cellular automata wizards Normal Margolus and Tommaso Toffoli for an article that ended up appearing in, disappointingly for me, a science fiction magazine.

In Beltsville, one of my hosts let me look through a glass window at an entire room full of computer chips, constituting a "Massively Parallel Processor" which was used to process terabyte arrays of satellite photo data, for instance removing stray "turd bits" by averaging pixels with their neighbors. I asked if he'd ever run the Game of Life on it, but he said no.

In today's paper, I see that the Lawrence Livermore labs are building a supercomputer called Thunder which will use several thousand Intel microprocessors in a parallel architecture. The machine is expected to run some ten thousand times as fast as an ordinary PC. Thunder won't quite match the power of today's fastest supercomputer, a Japanese machine called the Earth Simulator, which is used for simulations of climate change, such as global warming. America's Thunder machine, on the other hand, will be

[5] ibid, p. 22.

used, sigh, to simulate hydrogen bomb explosions.

### *Web as Mind*

Given that download/upload requests from the network nodes are, by and large, instigated by the machines' owners, there's a sense in which the Web is thinking society's thoughts. A widespread obsession with some topic reflects itself in the access patterns on the Web.

I'm not sure how far we can push this notion. Certainly an autonomously conscious Web isn't something that's happening right now. The situation might change if the component machines began exchanging information on their own. To some small extent this already happens. If an overzealous software vendor saddles you with an obnoxious "automatic upgrade" feature, your machine will make connections and sending and receive data without you asking it to. And certain viruses do the same. Here, again, my science-fictional mind is set to working. What if there were a computer worm or virus that served a higher purpose of awakening a complex computation across the Web? What if such a virus arose by accident? Or evolved?

People have tried to think about this prospect a little bit, but it's hard. I think of classic Hollywood-style science-fiction scenarios where scientists build a world-spanning computer and ask it, "Is there a God?" And the machine answers, "*Now* there is!" And then the newly roused computation sets to work kicking humanity's butt. But why? What would it have to gain?

After all, computers already dominate Earth. There's really nothing to overthrow, no power to seize. Our machines are the cells the planetary computer is made up of. And we devote considerable energy to building and maintaining these machines. We're *already* our computers' servants.

Computers wanting to kill humanity would make no more sense than, say, your brain telling the rest of your body, "All right, I'm going to kill all of you skin and muscle and bone and organ cells so that I can reign supreme!" Or, even crazier, your thoughts telling your brain, "Alright, I'm getting rid of all you lazy brain cells!"

This said, it is true that we try and encourage some parts of our body at the expense of others. We want more muscle and brain, less fat and tumors. Might the planetary Web mind decide to selectively eliminate certain elements? Indirectly this already happens. Web pages that use flawed or outdated code become obsolete and unvisited, because they don't support the evolution of the Web. Spammers get their accounts cancelled, not because of anything they stand for, but because they're bad for the efficiency of the Web. More of this wouldn't be such a bad thing. Imagine, say, the Terminator descending upon a trailer park of spammers in Boca Raton, Florida. ☺

### *Logistic Map*

As long as we keep A between 0 and 4, we can be sure that whenever x lies between 0 and 1, then NextValue(x) will lie between 0 and 1 as well.

### *Annealing Schedule*

To be quite concrete, suppose that you're searching a solution space made up of triples of real numbers, with all three coordinates ranging between minus one and plus one. You're searching through, in other words, a three-dimensional cube centered on the origin, and you want to find the "fittest" point. Suppose that you let the distance between two solutions be simply the usual Euclidean notion of distance. A conceivable "annealing schedule" might go like this.

- Take a population of 64 hill-climbers and set them down at random locations in the solution space. Set the temperature to 2.
- At each update, have each hill-climber evaluate the fitness at its present location and at eight randomly chosen locations that are within a distance of T from its current location. Move the hill-climber to the location among the nine points that has the highest fitness.
- Every ten updates reduce T by 0.1 until it reaches 0.1, and then begin reducing it by 0.01 every ten updates, and when it reaches 0.01 begin reducing it by 0.001 every ten updates.
- When T reaches 0.001 return the solution specified by the most successful of your hill-climbers.

Choosing the right annealing schedule for a given solution space is of course a metasearch problem of its own. Systematizing search processes is perhaps as much a craft as it is a science.

### *Self-Deprecating Remark Introducing Chapter 4*

When all else fails, systematize!

### *Continuous Response*

In the world of living organisms, we rather expect the inputs and outputs to be smooth, rather than abrupt zeroes and ones. If something is only a little bit hot, for instance, you might move your finger only a little bit.

### *Sigmoid Neurons*

There are various ways to set up these artificial neurons; I'll describe one commonly used approach. We'll suppose that one of our neurons can output a real number ranging from zero to one, and we'll suppose that it calculates this output by summing up some number of input signals with a particular weight being multiplied times each input signal. Unlike the signals, the weights are allowed to be negative, we let them lie between negative one and positive one.[6] We might think of negatively weighted inputs as inhibiting our artificial

---

[6] As always, when we talk about continuous numbers inside digital machines we need to pause and say that in practice the real numbers used by a PC aren't truly continuous, they're digital models of continuous numbers. As mentioned before, if we take the common expedient of representing a real number by thirty-two bits, then we're really allowing for "only" four billion different real numbers.

I also should mention that in practice, biological neurons do not have continuous-valued outputs. A brain neuron either fires or it doesn't fire; its output is more like a single-bit 0 or 1. Computer scientists prefer model neurons with continuous-valued ouputs because it's easier to tailor a smallish network of output-valued-output neurons to perform a task. If you insist on binary-valued-neurons, and positive values as activating them.

Given that we'll often start out with random weight values in our neurons, and that some neurons have quite a few input lines, it can well happen that in some cases the weighted sum can be a fairly large positive or negative number. In order to keep the output value in range between negative one to positive one, we "squash" the sum to a value between zero and one as indicated in the image of the artificial neuron.

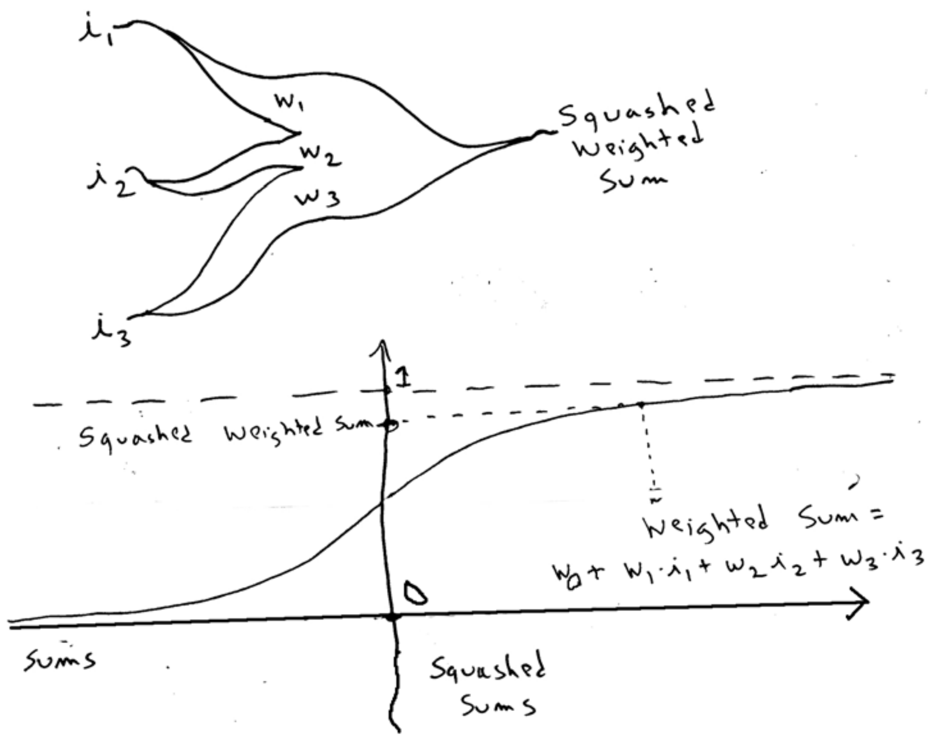



**Figure: An Artificial Neuron**

*A neuron computes the weighted sum of its inputs and uses this sum to calculate its output. Each neuron also has a "threshold weight" $w_0$ that it adds into the weight sum as well. In order to keep the output signal in the zero-to-one range, the neuron "squashes" the weighted sum before sending the result as an output..*

***My Anti-Beige Agenda***

It's possible that electronic computers are only a passing fad.

***Brain Surgery***

In a human being, changing the microcode is analogous to undergoing brain surgery or (in a temporary way) dropping acid; flashing the BIOS is akin to having a peak conversion experience; and changing the operating system is comparable to adopting a behavior-

---

output neurons, the networks become bigger and harder to train.

modifying regimen such as meditation classes or group therapy. A pathological neurosis can be so deeply ingrained a system bug that one has to dig down quite far to change it.

### *Good News Bad News*

Depending on how you look at it, this may seem like either good news (we won't be replaced by smart robots) or bad news (we won't figure out how to build smart robots).

### *Back Propagation as a Computation*

In a way, a neural net is something very simple. The net-training process itself has a simple and deterministic description: pick a network architecture, pick a pseudorandomizer and assign starting weights, and do back-propagation training on a sample set. The only complicated part here is the details of what lies in the sample set examples.

***

Again thinking back to evolution, you might wonder why we don't tune a neural net by using a genetic algorithm approach. You could look at a whole bunch of randomly selected weight sets, repeatedly replacing the less successful weight sets by mutations and combinations of the more successful weight sets.

You could do this, but in practice the neural net fitness landscapes are smooth enough that the back-propagation hill-climbing method works quite well — and its faster and simpler. But what about local maxima? Mightn't a hill-climbing method end up on the top of a foothill rather than on the top of a mountain? In practice this tends not to happen, largely because so many weights are involved in a typical real-world neural net. Each additional weight to tweak adds another dimension to the fitness landscape, and these extra dimensions can act like "ridges" that lead towards the sought-for global maximum, sloping up from a location that's maximal for some but not all of the weight dimensions. Also, as mentioned in the last chapter, we don't really need absolute optimality. Reasonably good performance is often enough.

### *Technical Problem with Running BZ on a Network Instead of on a CA*

A network doesn't have the neighborhood structure of a CA, like if A connects to B and A connects to C, that doesn't imply B is "near" C on a network, though it does in space? Well, actually, if A will promise to act as a relay station, then B and C would be near in that sense.

To go more towards network, how about giving each cell A its nearest neighbors NA plus one remote neighbor rA. And say we pick the remote neighbors so they aren't near each other. That is, if A and B are close then rA and rB aren't close. Maybe there could be a nice canonical way to do this. If I wanted to program this, I could have a CA with an extra real number field dist, and it would take the usual nearest neighbor inputs plus an input from a cell whose index is dist*N where N is the size of the CA array. I could fill the dist fields at startup with a randomizer. Oh, I did this years ago with John Walker, when I wrote the ZipZap and XipXap series of CAs in assembly language. The remote neighbors didn't change much.

### *Back-propagation Momentum Term*

As I mentioned in passing in the last chapter, when training a neural net, we also avoid getting stuck on local maxima by using a so-called momentum term which keeps us moving forward through the fitness landscape even when a maximum is found. If no better maximum occurs nearby, the momentum peters out and the search circles back.

### *Game AI*

There's a variety of ways one might approach this kind of problem. The so-called expert system approach is to analyze how a good football player avoids tacklers — when to fake, when to speed up, when to slow down, when to double back, and so on. With enough trial and error, you can code up an intricate reflex schema as a series of if-then rules, perhaps of the form:

**if Condition1 then do Action1;**
**if Condition2 then do Action2;**
**if Condition3 then do Action3;**
**if Condition4 then do Action4;**
**...**
**if Condition5 then do Action5;**

With sufficient care, you could make the conditions both mutually exclusive and complete — that is, you could define the conditions so that at any given time exactly one of them would hold. But this isn't always the best way to go. At some point it might be although **Condition1** is true, **Condition2** is very nearly true, so that the best thing might be to mix a bit of reaction **Action2** in with your reaction **Action1**.

How would I mix two actions? Some kinds of actions lend themselves to being mixed. If Action1 is to turn left by ten degrees, and Action2 is to turn left by thirty degrees, then one might possibly combine the two by turning left twenty degrees. But this could be a mistake. If, say, Action1 is to turn left ten degrees and Action2 is to turn right ten degrees, does combining them mean that you shouldn't turn at all? In some situations this might be exactly the wrong move. A different way to combine actions is probabilistically, as if by rolling weighted die. This is in fact the only option if the actions are incompatible, as when, for instance, Action1 is passing the ball and Action2 means kicking the ball.

To implement a mixed approach of any kind, rather than assuming that each condition is definitely true or false, we can instead give the truth of each condition a weight between 0 and 1. 0 is false, 1 is true, 0.2 is almost false, 0.9 is very nearly true, and so on.

This sounds nebulous, but we can make it precise. Suppose, for instance, **Condition** expresses the notion "the left guard is too close to me". If I let **distance** stand for the distance between me and the opposing team's left guard, then I might be comfortable in giving **Condition** a value of 0 if distance is greater than 10 yards, and giving **Condition** a value of 1 if distance is less than 1 yard. And maybe for the in-between distances I could use a linear expression of the form **(1 - distance/10)**, or if I preferred, a more complicated expression of like **(1 - distance2/100)**

Once we got the truth values for the various conditions we could use them in one of two ways, as coefficients in a weighted sum or as probabilities in a random die.

### *Incomprehensible Solutions*

The neural net for recognizing smiles and frowns, for instance, is just a mound of some three thousand real-valued numerical weights. You might hope that if you analyzed the weights, you'd discover that each of the hidden layer neurons is in fact learning to recognize a specific aspect of facial expressions. Perhaps the first neuron notices whether the left corner of the mouth bends down or up, perhaps the second focuses on the wrinkles or lack thereof in the face's brow, and maybe the third pays attention to the twist of the mouth's right corner. But this isn't necessarily the case. In many neural nets there's no easy way to pick out what it is that each individual hidden-layer neuron is learning. Indeed, the more compact and effective a neural net becomes, the more opaque it becomes, and the less amenable to human understanding. [Actually I don't think this is usually true.]

### *Unsupervised Learning Raps*

Let's see what happens if we try to think of the human brain as being like a computer scientist's neural networks. One immediate thing to note is that, because a human is embedded in the world, their brain network parameters are iteratively refined over the course of years and years of parallel computational effort. Think of the finger movements of a master musician or, for that matter, the empathetic understanding of an experienced teacher or counselor.

It's useful here to distinguish between two kinds of learning: supervised and unsupervised.

Supervised learning is like when I train a neural net with a set of a hundred test faces. I immediately tell the network which answers it got right and which it got wrong, and the network back-propagates the errors to carry out a hill-climbing process to adjust its three thousand or so internal weights. Supervised learning is like when you're learning addition in elementary school, and if you say "Five plus four is eight," the teacher right away says, "No, five plus four is nine," and perhaps you internally adjust on some neural connections in your brain.

But most of our learning is unsupervised in that the feedback is less immediate and direct.

***

Unsupervised learning is when K goes out in the playground and a girl won't get off the seesaw to give him a turn, and K bites her on the butt, but she's wearing thick snow pants and doesn't notice, but even so K feels bad and dumb. Unsupervised learning is when K looks for someone to marry and ends up with the girl he tried to bite in first grade. Unsupervised learning is when K finally tells her that old story and she divorces him. Error! Unsupervised learning is when L drinks so much beer at age sixteen that she throws up, learns to drink slowly enough not to get sick, but then realizes after thirty years of increasingly vexed imbibing that she suffers from alcoholism. Unsupervised learning is M dying of cancer from smoking cigarettes. Back-propagate that! Unsupervised learning is N laboring in obscurity on a novel for fifteen years and then selling it and having a nice success. N was right all along!

### *Exhortatory Slobber to Cover Up Intellectual Bankruptcy*

Even *without* external input, the mind's evolving computations are intricate and unpredictable, but we *do* have input, lots of it, the whole kaleidoscope of daily life. It's fun. Savor it.

### *Two-Year-Old Rebellion*

Developmentally, it may be that a baby first notices distinct objects in the world, and only a bit later "realizes" that he or she is a separate object on a par with those other things. Indeed, it's often said that this discovery is the essence of what stokes a two-year-old's rebelliousness. "Mommy and Daddy are different from me. I don't have to do what they say. If I yell, maybe they'll do what *I* say."

### *Mercury Delay Lines*

The trick used for memory storage in the earliest electronic computers was almost unbelievably strange, and is no longer widely remembered: bits of information were stored as sound waves in tanks of liquid mercury. These tanks or tubes were also called "mercury delay lines." A typical mercury tube was about three feet long and an inch in diameter, with a piezoelectric crystal attached to each end. If you apply an oscillating electrical current to a piezoelectric crystal it will vibrate; conversely, if you mechanically vibrate one of these crystals it will emit an oscillating electrical current. The idea was to convert a sequence of zeroes and ones into electrical oscillations, feed this signal to the near end of a mercury delay line, let the vibrations move through the mercury, have the vibrations create an electrical oscillation coming out of the far end of the mercury delay line, amplify this slightly weakened signal, perhaps read off the zeroes and ones, and then, presuming that continued storage was desired, feed the signal back into the near end of the mercury delay line. The far end was made energy-absorbent so as not to echo the vibrations back towards the near end.

How many bits could a mercury tube hold? The speed of sound (or vibrations) in mercury is roughly a thousand meters per second, so it takes about one thousandth of a second to travel the length of a one meter mercury tube. By making the vibration pulses one millionth of a second long, it was possible to send off about a thousand bits from the near end of a mercury tank before they started arriving at the far end (there to be amplified and sent back through a wire to the near end). In other words, this circuitry-wrapped cylinder of mercury could remember 1000 bits, or about 128 bytes. Today, of course, it's no big deal for a memory chip the size of your fingernail to hold billions of bytes.

### *Robot Emotions*

It's worth mentioning that the roboticist Rodney Brooks thinks of machine emotions in quite a different way. For Brooks, emotions are to be contrasted with reason. He feels that we're looking at artificial reasoning when a robot or program arrives at a decision by means of some kind of internal simulation of the situation, possibly combined with a logical analysis. And for Brooks, a robot emotion is when a machine simply responds, as when using a built-in reflex.

### *Nonlinear Brain Waves*

The glider model is somewhat more appropriate than waves because we don't in fact see our thought trains spreading out and diffusing, they tend to stay rather narrowly focused. But if we really want a wave-like model of thought trains, we can in fact represent narrow, focused waves by something called non-linear solitons.

### *Brain Neuron Rules?*

If the brain's activity is something like a CA running on a network of neurons, we can ask what is the rule in each neuron. One might use a complicated or a simple idea. The complicated idea is to have different rules at each neuron. The simple idea is to have the same rule at each neuron.

### *Consciousness Isn't Hardware or Software*

The curious thing is that I can imagine changing any component of my hardware or software without actually affecting my essential sense of being me. My body's cells are always replacing themselves. And I'm always forgetting old events and learning new things.

***

The notion that your consciousness is separate from your hardware and software fits with the sense that you can change a great without affecting your essential sense of personal identity. Your body's cells are always replacing themselves, for instance, and you're always forgetting old events and learning new things.

### *Extra Damasio Quote*

...the neurobiology of consciousness faces two problems: the problem of how the movie-in-the-brain is generated, and the problem of how the brain also generates the sense that there is an owner and observer for that movie. ... In effect, the second problem is that of generating the appearance of an owner and observer for the movie within the movie. ( page 11 or 111).

### *Game creature psychology*

Speaking more formally, we give the creature an integer variable called **recent_success** which keeps a running total of how much the creature's score has changed over, say, the last hundred updates. And now suppose that the creature has access to several alternative **feeling_array** options. And, finally, suppose that the creature's **update** method does the following:

(i) recompute the rolling **recent_success** indicator to take into account the score change, if any, produced by the most recent move;

(ii) possibly change the values stored in **feeling_array** if **recent_success** is deemed to have too low a value;

(iii) update the creature's current motions on the basis of the currently active **feeling_array** as applied to the other creatures' locations and types.

### ***Snail and dog consciousness***

Immersion might be thought of as experiencing the world with an empty mind, no model of it needed, no image, no notion of objects, simply the world in and of itself, letting "the world think me" instead of "me thinking the world."

A snail doesn't even have Boundaries. I'm not sure if a dog has a movie-in-the-brain or not, maybe only fleetingly. Damasio talks about some brain-damaged people who have the movie-in-the-grain but not feelings and core consciousness.

### ***Magpie***

The magpie approach to writing. Stick in any old thing you've seen and see what grows out of it. Like seeding a solution for crystals.

### ***Fruits of Reason***

3D space to fit objects into.

Time with before and after.

Grammar. Grammar as logic-like tool for producing coherent new utterances.

Fun to laugh at how dumb dogs are. But sad, too. Poor Slug squeezed up against the door, listening to our voices.

Visual perception important: feedback, robot arm uses to pick things up.

Finding a location: precalcuating the route or repeatedly asking directions.

Music.

Grammar(Many, Objects), Logic(Spacetime, prognostication), 3D Perception (One, World)

### ***Class 3 Thoughts?***

I think it's very rare we have Class 3 thought patterns. A chaotic storm. Like maybe when I took acid it was like that? Not even. That was more like Class 4 very dense and fast. Though maybe the White Light, or any milder type rush is class 3, with all the neurons firing there's no real content, simply a sense of stimulation. Any connection here to why stupid people (athletes, businessmen) like coke?

### ***Comparing Gibson to a Lifebox***

Talking to my famous writer friend William Gibson is a little like talking to a lifebox of him. My guess is that Bill's been interviewed so much that, unless I say something very personal or unusual to him, he tends to respond with a mentally taped answer he's used before. I could be wrong about this, but his answers are so witty and articulate that I don't see how he could possibly be making them up on the spot.

### ***Wattahertz Evolution***

| **Quantity** | **Estimate in words** | **Power of ten** |
|---|---|---|

| | | |
|---|---|---|
| Population | A million individuals to get evolution. This means a million at any given time, not a million in all. On the low side, but remember that the population used to be much smaller. | $10^6$ |
| Number of years of evolution | A million. Again, this is somewhat on the low side, but most of the important brain features may have evolved fairly early. | $10^6$ |
| Seconds per year | Thirty million, approximately. | $3 * 10^7$ |
| Instructions per brain per second | Three hundred quadrillion, as calculated in the previous table. | $3 * 10^{17}$ |
| Total number of instructions processed to simulate human evolution | Population * years of evolution * seconds per year * instructions per brain per second = $3*3*10^{(6+6+7+17)} = 10^{37}$ updates if again I round 3*3 up to 10. A mathematician would use the name ten duodecillion for $10^{37}$. | $10^{37}$ |
| Instructions-per-second needed to simulate evolution in a year | Total number of instructions / seconds per year = $10^{37}/ (3*10^7) = 3 * 10$^$^{29}$. Here I'm rounding 10/3 down to 3. | $3 * 10^{29}$ |
| Target clock rate | If there's three clock ticks per machine instruction, we get $3 * 3 * 10^{29}$, which we round up to $10^{30}$, a tidy million yottahertz, which could also be called a wattahertz. | $10^{30}$ |

**Table: A One-Year Simulation of Human Evolution on a Wattahertz Machine**

### *Why I Quit Teaching*

I was tired of preparing new lectures and demos on difficult material, tired of wrestling with the ever changing hardware and software, and eager to devote my few remaining non-senile years to writing.

### *Freewill*

Once the dust settles, what you did *is* what you wanted to do. This truism lies at the core of the philosophy called existentialism: *you are what you do*.

### *Recovery-Group-Tinged Rap About God and Free Will*

Once or twice a week I participate in a meeting of a support group devoted to helping people change their destructive behaviors. One of the pleasant side-effects of this kind of informal group therapy is that I get flashes of insight into the minds of people whom I might otherwise never meet. Over and over I discover unexpected intricacies of emotion and humor within strangers. Nobody is simple on the inside. It's an impossibility. Every brain is carrying out a class four computation.

And this is no surprise, really. For look inward at your flow of thought. It's like that cellular automaton rule depicted above. One thing leads to another. The glider-like thought trains collide and light up fresh associations. Even if you're lying in bed with your eyes closed, the flow continues, the endless torrent. Now and then you get stuck in a loop, but some unexpected glider eventually crashes in to break things up. You're surfing the brain waves; and you yourself are the surf.

Backing up, what about those meetings I go to, huh? By the time I turned fifty, I'd developed some self-destructive habits that I wanted to stop; that's why I go to meetings. I'm a new man. Is this, *aha!*, an example of will power?

My support group espouses a deliberately paradoxical attitude towards will power. The way for me to change was to recognize that, on my own, I was unable to change. So how did I change? Well, I jumped out of the loop and asked God to help me.

Does that mean that I think the Creator of the Universe, the Ground of All Being, the Omnipresent-Omnipotent-Omniscient One has reached down to poke into my brain and change the parameters of Rudy Rucker's mental computation? Maybe. An All-Powerful God would, after all, have enough time and energy to get around to even the smallest tasks.

And why shouldn't there be a God — whatever that means? At the very least, one can take "God" as a convenient and colorful synonym for "the cosmos".

Having survived six early years in the Lynchburg, Virginia, of Jerry Falwell's so-called Moral Majority, I do take religion with a grain of salt. A less supernatural view of my reform would be to say that my asking God to help me has an organic effect upon my brain's computation. Expressing a desire to have a spiritual life activates, let us say, certain brain centers which release endorphins that in turn affect the threshold levels of one's neurons.

Do I really think it works like that? Well, to be truthful, I've always felt comfortable about reaching out for contact with the divine. The world is big and strange, and we have only the barest inkling about what lies beneath the surface.

Maybe the cosmos is dancing with us all the time. Maybe God is in the blank spaces between our thoughts, like in those white regions of the picture of the China CA.

Anything's possible.

### *Enlightenment Is...*

Enlightenment is when you let the world think you — instead of you thinking the world.

### *Computers Don't Get Bored*

One seeming difference between humans and personal computers is that that the machines don't get bored and give up in disgust when you ask them to do something very tedious and repetitive. Our machines seem to excel at being stupid really fast. Computing a fractal image, for instance, requires carrying out several hundred multiplications for each pixel in the image. No human would ever carry out all these steps. It wouldn't be feasible.

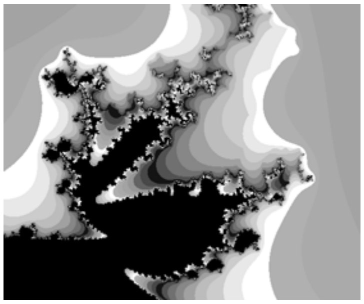


**Figure: Detail of a Cubic Fractal Akin to the Mandelbrot Set.**

But wait. Humans do carry out immense amounts of repetitive information processing. Consider vision, for instance. As you walk around, your brain is effortlessly crunches huge amounts of data from your retinas' rods and cones. You pick out lines and surfaces, you fit the surfaces together into three-dimensional shapes, you assemble the shapes into coherent models of the world around you. You do all of this unconsciously, and it doesn't strike you boring at all. Far from it! One of the particular treats of tourism is looking at unusual three-dimensional objects: cathedrals, reefs, ballparks, forests, skyscrapers, canyons and the like

One also tends to think that personal computers have exceedingly good powers of data retrieval. If you open the file of a book-length manuscript, the machine has no problem in searching through the whole file and replacing, say, every occurrence of "towards" with "toward." For a person this is feasible, but somewhat tedious and time-consuming.

But again, we humans have incredibly powerful data retrieval skills as well. It's just that our brains haven't evolved for the kinds of tasks that the machines excel at. You can see an object like a spatula from any angle and immediately know what it is. You need only glimpse part of a person's face to recognize the individual. On a good day, in a spirited conversation, the words leap from your tongue tip; you call up apropos phrases drawn from the great body of media you've taken in.

### *Word Virus*

A successful creative artist parasitizes people's minds. Burroughs: the word is a virus.

### *The Use of Logic*

Logic is a tool for reaching beyond your simulations. Logic unearths new facts that you can't see, or that haven't happened yet.

### *Sand and Books*

Books and art works, after all, aren't indistinguishable grains of sand, and people aren't identical cells that always react to inputs in exactly the same way.

But most books are like feathers dropped into the Grand Canyon.

### *Caption for Photo with Nick Herbert*

*Yes, I'm wearing lipstick. It was just to look weird for the parade.*

### *Rant at Start of Chapter on Society*

I write this book during a dark time. America's government is in the hands of a ruthless antidemocratic elite.

But history is not about politicians. A baby filling a diaper is infinitely more significant than a congress placing a movement on the floor.

### *Joy of Hacking*

Coding a simulation forces a programmer to ponder unexpectedly many issues. God is in the details. One might go so far as to assert that a person doesn't fully understand something until they've written a simulation of it — a precept which has the perhaps too übergeekly corollary that non-programmers don't fully understand *anything*!

### *Peace March Safety Example of Enjoying a Crowd*

During the Vietnam War, and again during the second Iraqi War, the only times I felt truly safe from my nation's government was when I was part of a protest march.

### *Games and Flocking*

I think it's a shame that online massively multiplayer computer games don't presently take into account the joy of flocking. In all too many games, the only interaction you have with people is to attack them with a weapon, to run away from them, or, at best, to share a ride in a vehicle. Games are more interesting and humane if a player has to dance with or walk around with the other characters — instead of simply killing them. Sports games are something of an exception, with, for instance, open field running in a football game being a nice example of crowd-motion play.

### *Self Reference*

(I'd inserted this remark in the middle of my discussion of hive mind consciousness) Here the discussion becomes more class four, more a matter of me making things up as I go along.

### *I Want More Influence*

Yes, I've had some slight effect upon my hive by voting in elections, by discussing my opinions with people, by raising children, by teaching classes, and by publishing books. I should be satisfied with this, but at times I'm not.

***

I only harp on the problem of geezerly obsession with the national hive mind because I can see my own tendencies in this direction — and I'm sure you can, too, given that I've been not-so-subtly criticizing our media for couple of pages now. Ain't it awful?

***Twin Towers***

Four Facts: The twin towers fell. The terrorists were *Saudis*. Bush invaded *Iraq*. Most people feel Bush made a wrong decision.

"Ah," someone might say, "if nobody wanted to fight, we'd be invaded. Look at the twin towers. The world's not safe." And I would submit that the administration's reaction to the twin towers was exactly the wrong one. Instead of jumping into the repetitive tit-for-tat class two Israelis-versus-Palestinians mode, the government should have gone class four. What would make men kill themselves while destroying a part of our lovely New York City? What system produced them? Isn't there a way to get in and jolt it in some totally unexpected way, something more original than rocket fire vs. car bombs?

***Emigration***

Before virtually every American presidential election, I've heard people say, "If so and so wins, I'm leaving the country." But they never do. The only time my friends ever emigrated was during the Viet Nam war, a time when the hive mind was undertaking the wholesale slaughter of a generation. But most of the time, for most of us, things aren't bad enough to make emigration seem reasonable.

If national elections were to be stolen over and over again, the answer might be armed revolution, not emigration. At some point, a significant number of people might feel compelled to go to D.C. and fight in the streets until the regime were to be deposed. However long it took them, however dearly it cost.

***Artificial Life***

Machines will never be alive — because they aren't born with cunt juice on them.

***The multiversal web.***

The parallel worlds of the quantum multiverse are in many senses the same as web pages. Can I use the web as a model for the universe itself?

***I'm a Rebel***

I'm congenitally rebellious, prone to opposing every form of group opinion. Seeds for this attitude would be that I grew up as a younger brother, that I was youngest and most intelligent boy in my grade school classes, that I barely escaped being sent to die in the government's war in Vietnam, and that for many years I was a pot-smoker and thus a criminal.

My feeling is that essentially everything in the newspapers or in the history books is a trick, a lie, and not really worth talking about. The survival strategy of some elements of society consists of a two-step process:

- Frighten people.
- Sell them protection.

### *Losing the* **Search for Infinity** *Deal*

We got close, but we didn't quite make it, possibly because Mandelbrot himself put in a word against the project, or possibly because I made the error of sending Jeff my twenty-page treatment in Microsoft Word format rather than in Acrobat PDF format. Jeff's version of Word had a different normal.dot file, which removed all the breaks or indentations between paragraphs — turning my eleven-draft twenty-page script into, sob, a repellently monolithic block of text. And Jeff, who'd waited till the very last hour to send in the proposal, didn't notice before emailing it in. Whatever.

### *Extra Rows for the Excitation Table*

| | | | | |
|---|---|---|---|---|
| **Replenishing Rule** | Randomly change some 2 → 0 | Always 2 → 0 | People become ready for something new | Resting neurons recover and become ready to fire |
| **Seeding Rule** | Randomly change some 0 → 1 | Only seed at startup | Artifacts are launched | External inputs to the brain |
| **Spreading Rule** | 0 → 1 if any neighbor is 1 | 0 → 1 if two neighbors are 1 | People tell their friends about an artifact | Neurons stimulate their neighbors |
| **Exhaustion Rule** | Always 1→ 2 | Always 1→2 | People lose interest in an artifact | After firing, a neuron needs to rest |

### *The Starwars CA*

As a visually-oriented cellular automatist (i.e. a computer fanatic who spends hours staring at weird screens), I tend to be more interested in dynamic behaviors than in statistics. I'm more intrigued by scuttling gliders and writhing Zhabotinsky scrolls than I am by log log graphs of power laws. Might we hope to model social behaviors by fully deterministic CAs that seethe interestingly when started with pretty much any kind of pattern at all?

Certainly Brian's Brain is a class four computation capable of producing, all on its own, as much disorder and gnarl as we need — and it never dies out. And there are many rules like it. In the figure below I show an excitation-based rule called StarWars that settles into a pattern of grid-like globs with sparks racing around the glob edges and gliders shuttling back and forth.

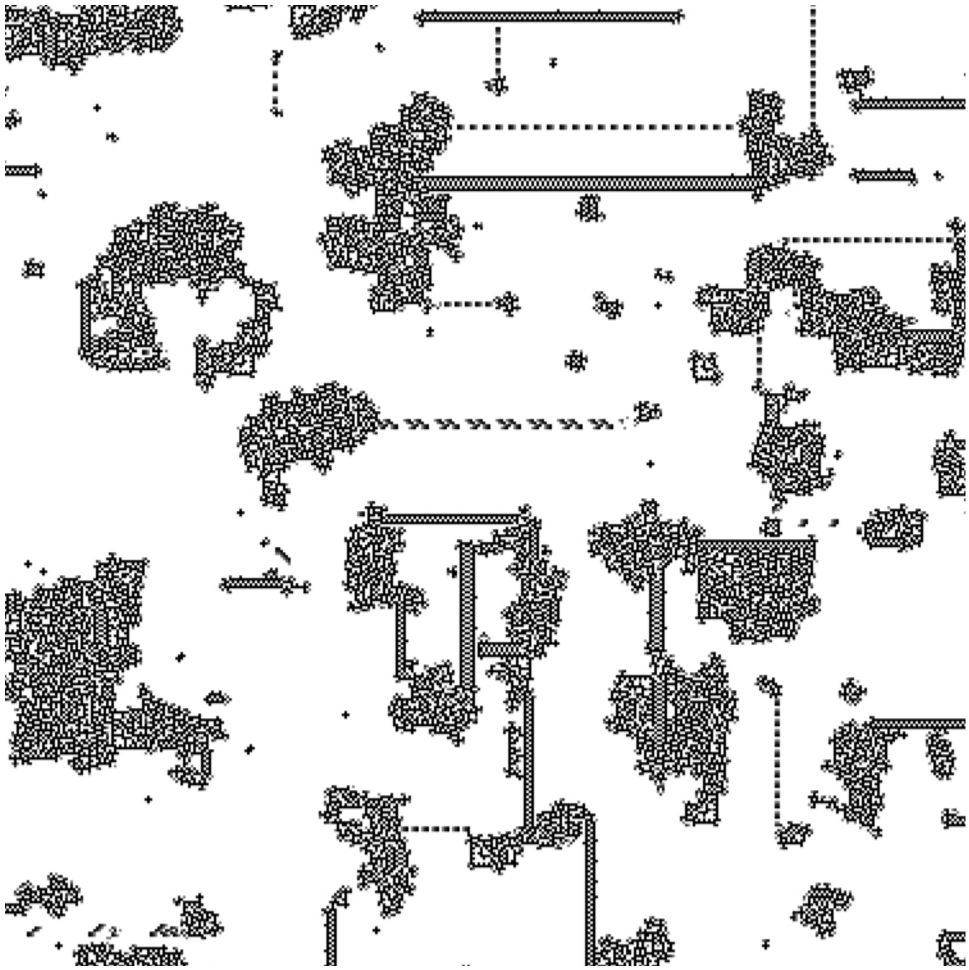


**Figure: The StarWars Rule.**

*Starting from a random bath of pixels, StarWars settles into pattern resembling an aerial view of a busy factory. In the center we have a glob shooting out a regular stream of gliders which disintegrate when they hit the little block to its right. The more regular-looking horizontal and vertical lines are also streams of gliders. Small sparks flicker around the edges of the globs. Now and then a rogue glider crashes into a glob and kicks up a prolonged cascade of reactions.*

The StarWars rule has four states, which we think of as being a ready state, a firing state, and two distinct two resting states. The update rule is similar to that of Brian's Brain, with two differences: it's possible for a cell to continue firing for more than one update, and the cells rest for two updates instead of one. The grids are made up of firing cells that stimulate each other to continue firing, while the little gliders are made of firing cells trailed by resting cells.

StarWars worlds smaller than 500 by 500 pixels become periodic in about 40,000 generations, but I'm not sure if the larger worlds begin repeating themselves within any reasonable time frame. At the very least the rule is class two, but my feeling is that, given how easy it would be to build a virtual computer out of a StarWars pattern, it is reasonable to call it class four. It's also my impression that if you click one of the dead pixels into the firing state, the effects may propagate outwards to an arbitrarily large degree. It could be that the cascade sizes obey a histogram power law, in which case StarWars would serve as a class four rule which supports inverse power law statistics.

The StarWars image was made with Mirek Wójtowicz's downloadable Windows executable *Mirek's Cellebration*, which can be obtained from his http://www.mirekw.com/ . Note that Mirek has a Java version of his program online as well, although the standalone executable seems to run a bit faster.

Mirek discovered the StarWars rule. It has 4 states which we think of as being the ready state 0, the firing state 1, and the 2 resting states 2 and 3. The update rule is as follows.

If a cell in the ready state 0 has exactly two firing neighbors it goes to the firing state 1.

If a cell in the firing state 1 has three, four or five firing neighbors then it stays in state 2. Otherwise it goes to the resting state 2.

A cell in the resting state 2 goes to the resting state 3.

A cell in the resting state 3 goes to the ready state 0.

### *Examples of News Stories*

Nearly everyone knows about the big ones. The invention of automobiles. The Second World War. The fall of the Berlin Wall. The dot com bubble. Harry Potter. 9/11.

### *Life is Hard at Every Level*

Returning once again to my concern about being a mid-list non-best-selling author — here I am stuck out on the fat-but-not-all-*that*-fat tail of the gravy train.

Criticality means that at least things are boiling at every level. So it's never going to be dull. In fact it's not easier to send your stories to very low-circulation magazines or take jobs you're over-qualified for. There's just as much hassle at every scale where you're competent.

### *Power Law Notes*

Natural phenomena often obey a certain scaling property called a power-law distribution. To set the stage, we first break some phenomenon into components of various sizes, and then for each size level, we evaluate how much the components of this size contribute to the phenomenon as a whole. In a power-law distribution, there is some characteristic exponent D such that the relative contribution of pieces of a given size R is proportional to the reciprocal of R raised to the power D.

**Intensity of contribution from the pieces of a given size $R \sim 1 / R^D$**

More formally, there will an exponent D and some constant factor C such that if P(R) is the intensity or strength of the contribution from the pieces of size R, then

$$P(R) = C \;/\; R^D$$

Let me illustrate with a simple example, the sizes of rocks on a mountainside, which obey a power-law distribution with D = 1. To begin with, we'll measure the sizes of the rocks in terms of their volume. And we'll measure the "intensity" of the rocks of a given size by counting how many rocks of that type there are.

For simplicity's sake, I'll describe a way of getting rocks of different size levels by repeatedly making copies of the existing rocks and break the copies in two.

- *Level 0*. Start with a big rock of size 1.
- *Level L+1*. Make copies of the Stage L rocks and break each of the copies into two equal pieces.

Admittedly, the notion of copying rocks is artificial. But as you can see in the figure below, we could also think in terms of starting with an appropriate number of big rocks and breaking each of down to a different level.

*[Figure (not drawn)] We can reach L levels of division by starting with L+1 big rocks and shattering each of them through successively more steps.*

If we go down to level L, we then have:

- 1 piece of volume 1,
- 2 pieces of volume 1/2,
- 4 pieces of volume 1/4 ... and
- $2^L$ pieces of volume $1 / 2^L$.

Now in each row, the volume of the pieces is the quantity $R = 1/ 2^L$ on the right, and the number of pieces of that volume is the quantity $P(R) = 2^L$ on the left. If I combine the two equations I get:

$P(R) = 1 / R$

So in this case, as promised, we have a power law with D = 1.

Now let's ring in a change. Suppose we think of the rocks as cubes, and that we measure their size in terms of their edge length instead of their volume? And let's also suppose that when we break a cube in two, we can, without changing the total volume, knead the pieces like clay so that they too are little cubes. In this case, we'd have:

- 1 piece of edge 1,
- 2 pieces with an edge equal to the cube root of 1/2,
- 4 pieces with an edge equal to the cube root of 1/4 ... and
- $2^L$ pieces with an edge equal to the cube root of $1 / 2^L$.

If we write R for the edge length, then at level L, $R^3$ equals $1 / 2^L$, and the reciprocal of this is the same as the intensity $P(R) = 2^L$. So we get this kind of power law.

$P(R) = 1 / R^3$

If you don't like the idea of kneading the pieces into cubes, you could instead be cutting each cube into eight pieces.

1 piece of edge 1,
8 pieces of edge 1/2,
64 pieces of edge 1/4 ... and
$8^L$ pieces of edge $1 / 2^L$.

Now here, since $8^L$ is the same as $(2^L)^3$, we again have

$P(R) = 1 / R^3$

These exponent 1 and exponent 3 descriptions of the sizes of rocks become rather similar if we graph them in a certain way. The idea is to use log-log graph paper.

A different type of power law arises with natural sounds such as the babbling of a brook, the whispering of the wind in the trees, the crashing of the ocean, or the splashing of rain. We can view these sounds as being made up of different-sized pieces in the sense that they're combinations of waves of different frequencies or cycles per second. The deep sounds have low frequencies and the shrill sounds have high wavelengths. Here we view the "size" of a wave as being its frequency, and we view its intensity as the amplitude of the sound's contribution. Now it so happens that most natural sounds obey the power law.

Sound component amplitude = 1 / frequency

Fractals can be characterized as systems that obey power laws with fractional exponents. The Koch curve we can measure the size of a piece by its linear length, and the intensity of that size by the number of components of that size. It turns out that we get levels like this, but with the levels overlaid upon each other.

1 piece of length 1,
4 pieces of volume 1/3,
16 pieces of volume 1/9 ... and
$4^L$ pieces of volume $1 / 3^L$

Now it turns out that if D = log(4)/log(3), then $3^D = 4$.

The nineteenth century Italian economist Vilfredo Pareto observed a kind of power law in income distribution, with people's income being proportional to the inverse of some power of their rank. He came up with a simple kind of model to suggest how this could come about.

In the more general case, suppose that each employee has K subordinates, and each subordinate earns 1/L as much as his or her immediate boss.

If we generate the organizational chart down to N levels, viewing level 0 as being the single supreme boss at the top, then the people at the Nth level will be getting a salary S of $(1/L)^N = (1/L^N)$ times the supreme boss's salary.

The number of people at the Nth level will be $K^N$. Their rank R will also be on the order of $K^N$, because if we sum up the rows of employees ranking higher than the Nth level people, we are looking at the sum with i ranging from 0 to N-1 of $K^i$, which is $(K^N - 1) / K-1$, a quantity which is roughly $K^{N-1}$ at the beginning of the row, but $K^N$ at the end of the row.

Now define D to be (log L) / (log K), then $K^D$ is L and if we say R is $K^N$, then $R^D = (K^N)^D = L^N$.

Then $S = 1/L^N = 1/(K^N)^D \sim 1 / R^D$ or, quite simply,
$S \sim 1 / R^D$.

### *Inverse Power Law Examples*

Inverse power laws are all over the place; here's a few more examples.

| **Category** | **Success** |
|---|---|
| Job | Salary |
| Company | Net Income |
| Author | Average book advance |
| Movie | Gross receipts on opening weekend |
| Artist | Average price paid for a work |
| Web page | Number of links to this page |
| City | Population |
| Person | Number of close acquaintances |

In each case we can draw a graph of how the success value drops off as we move from the higher ranking towards the lower ranking individuals in a class. As it turns out, these graphs tend to drop off more steeply than one might expect. They obey distribution rules known as power laws. Zipf's law is an especially simple power law. In the more general case, the rank may be raised to some power **D**. So we expect for each individual in the sample,

**Success measure of an individual I = C / (Success rank of the individual I)$^{D}$**

Now in many situations, it turns out not to be practical to rank the entire sample in order of success. It can be more useful to express a power law in terms of the probability of finding an individual of a certain level. In terms of salaries, for instance, we might take a large sample of people, determine their salaries, and determine the probability of finding a person at each salary level. In this case the power law could be written in this form.

**Success measure S = A / (Probability of finding an object with success measure S)$^{D}$**

The two kinds of power laws are roughly equivalent.

In order to graph these kinds of laws, scientists often use what's called logarithmic graph paper. That is, the scales along the two axes can be set to, say, powers of ten instead of measuring uniformly along the axis. If we do this we are in effect plotting log(y) against log(x), which has the effect of turning power-law graphs into downward-sloping straight lines which are more or less steep depending on the size of D.[7]

---

[7] Suppose I take my logarithms to base ten, and **u** and **v** are, respectively the **log** base ten of **y** and **x**. This means that $\mathbf{x = 10^{u}}$ and $\mathbf{y = 10^{v}}$. In this case, a power law of the form $\mathbf{y = C/x^{D}}$ has the form $\mathbf{y = x^{-D}}$ because a negative exponent means division.

We can convert this step by step to a linear equation giving **v** as a function of **u**.

$\mathbf{10^{v} = (C * 10^{u})^{-D}}$

$\mathbf{\log(10^{v}) = \log((C * 10^{u})^{-D})}$ because the logs of equals are still equal.

$\mathbf{\log(10^{v}) = \log(C) + -D * \log((10^{u}))}$ because logarithms turn products into sums and exponents into products.

$\mathbf{v = \log(C) - D*u}$ because the log base ten of $\mathbf{10^{a}}$ is always just **a**.

$\mathbf{v = B - D*u}$, writing **B** for the constant **log(C)**.

### *Zipf's Law Table*

The table below illustrates Zipf's law, based on a word sample consisting of some five hundred old articles from *Time* magazine totaling about a quarter of a million words.[8] The idea is that, row by row, the last two columns should be roughly equal to each other. [9]

[8] I found this data on a web page by computer scientist Jamie Callan of the University of Massachusetts, http://web.archive.org/web/20001005120011/hobart.cs.umass.edu/~allan/cs646-f97/char_of_text.html. For the mother of all Zipf's law web pages, see Wentian Li's site, http://linkage.rockefeller.edu/wli/zipf/.

[9] He scanned an XML edition of Webster's 1913 Revised Unabridged Dictionary, as revised and extended by the GNU Collaborative International Dictionary of English project, found at http://www.ibiblio.org/webster/

We defined two words to be linked if either appears in the definition of the other. Walker created a table of pairs (L, N) which states the number of words N having a given number of links L, with L from 1 to 149. There are three traditional ways of describing this kind of data.

An inverse *power law* of the form $N \sim 1/L^D$ . This means that the number of words N that have a given linkiness L is proportional to 1/ L.

**A** *Zipf style law* (to be discussed a bit later in this section) in which we rank words from the most linked to the least, and if R is a word's rank order, then the linkiness $L \sim 1 / R^E$.

A *Pareto style law* that would say that for any linkiness L, the number of words M having linkiness greater than L is $M \sim 1 / L^F$.

A paper by Lada A. Adamic, *Zipf, Power-laws, and Pareto - a Ranking Tutorial* at http://www.hpl.hp.com/research/idl/papers/ranking/ranking.html, points out that we can get from a power law to a Pareto style law to a Zipf style law and vice versa. Let's see how to go from our power law data to a Zipf form.

*Power*. I have $N = c/L^D$. L is a linkiness level and N is the number of words at that level.

*Pareto*. To get the number of words with linkiness greater than $L_0$, integrate $c*L^{-D}$ with respect to L from $L_0$ to infinity. Assume D > 1. I get $(c/(1-D))*L^{(1-D)}$ evaluated from $L_0$ to $\infty$, which cooks down to $(c/(D-1)) / L_0^{(D-1)}$. So I can say that if M is the number of words with linkiness greater than L, then $M = (c/(D-1)) / L^{(D-1)}$.

*Zipf*. If I rank words in order of linkiness, and R is the Rth ranking word and it has linkiness L, then all the R words of higher rank have linkiness better than the word in question, so in fact R is the same as the M of the Pareto form, so I can say $R = (c/(D-1)) / L^{(D-1)}$. And now if I turn this around to solve for L in terms of R, I get $L = e / R^{(1/(D-1))}$, where $e = (c/(D-1))^{(1/(D-1))}$.

Some *Mathematica* curve-fitting to Walker's data gave me these numbers:

*Power Law*. $N = 1{,}000{,}000 / L^{2.2}$.

*Pareto Style Law*. M is the number of words with linkiness above L, and $M = 833{,}333 / L^{1.1}$

*Zipf Style Law*. L is linkiness of the Rth ranking word and $L = 244{,}312 / R^{\wedge 0.91}$

| Word | Number of occurrences in a Quarter-Million-Word Sample | Rank | 0.1/Rank | Probability (= Occurances/250,000) |
|---|---|---|---|---|
| the | 15861 | 1 | 0.1000 | 0.0646 |
| of | 7239 | 2 | 0.0500 | 0.0295 |
| to | 6331 | 3 | 0.0333 | 0.0258 |
| a | 5878 | 4 | 0.0250 | 0.0240 |
| and | 5614 | 5 | 0.0200 | 0.0229 |
| in | 5294 | 6 | 0.0167 | 0.0216 |
| that | 2507 | 7 | 0.0143 | 0.0102 |
| for | 2228 | 8 | 0.0125 | 0.0091 |
| was | 2149 | 9 | 0.0111 | 0.0088 |
| with | 1839 | 10 | 0.0100 | 0.0075 |
| his | 1815 | 11 | 0.0091 | 0.0074 |
| is | 1810 | 12 | 0.0083 | 0.0074 |
| he | 1700 | 13 | 0.0077 | 0.0069 |
| as | 1581 | 14 | 0.0071 | 0.0064 |
| on | 1551 | 15 | 0.0067 | 0.0063 |
| by | 1467 | 16 | 0.0063 | 0.0060 |
| at | 1333 | 17 | 0.0059 | 0.0054 |
| it | 1290 | 18 | 0.0056 | 0.0053 |
| from | 1228 | 19 | 0.0053 | 0.0050 |
| but | 1138 | 20 | 0.0050 | 0.0046 |
| u | 955 | 21 | 0.0048 | 0.0039 |
| had | 940 | 22 | 0.0045 | 0.0038 |
| last | 930 | 23 | 0.0043 | 0.0038 |
| be | 915 | 24 | 0.0042 | 0.0037 |
| have | 914 | 25 | 0.0040 | 0.0037 |
| who | 894 | 26 | 0.0038 | 0.0036 |
| not | 882 | 27 | 0.0037 | 0.0036 |
| has | 880 | 28 | 0.0036 | 0.0036 |
| an | 873 | 29 | 0.0034 | 0.0036 |
| s | 865 | 30 | 0.0033 | 0.0035 |
| were | 848 | 31 | 0.0032 | 0.0035 |
| their | 815 | 32 | 0.0031 | 0.0033 |
| are | 812 | 33 | 0.0030 | 0.0033 |
| one | 811 | 34 | 0.0029 | 0.0033 |
| week | 793 | 35 | 0.0029 | 0.0032 |
| they | 697 | 36 | 0.0028 | 0.0028 |
| govern | 687 | 37 | 0.0027 | 0.0028 |
| all | 672 | 38 | 0.0026 | 0.0027 |
| year | 672 | 39 | 0.0026 | 0.0027 |
| its | 620 | 40 | 0.0025 | 0.0025 |
| britain | 89 | 41 | 0.0024 | 0.0004 |
| when | 579 | 42 | 0.0024 | 0.0024 |
| out | 577 | 43 | 0.0023 | 0.0024 |
| would | 577 | 44 | 0.0023 | 0.0024 |
| new | 572 | 45 | 0.0022 | 0.0023 |
| up | 559 | 46 | 0.0022 | 0.0023 |
| been | 554 | 47 | 0.0021 | 0.0023 |
| more | 540 | 48 | 0.0021 | 0.0022 |
| which | 539 | 49 | 0.0020 | 0.0022 |

into 518 50 0.00200.0021

**Table 5.1: The Probability of a Word is About 0.1 Divided by the Word's Rank.**

Since the base sample for this table is from a news magazine, there are many mentions of the U. S., which is why we see “u” and “s” as frequent words.

***Web Hype***

The Web provides us with an unprecedented level of knowledge amplification; in effect we have access to a global mind.

***Memes***

One final thought. It could be fruitful to think of cultural artifacts as independent beings that travel about in the hive mind. Mind parasites, as it were. The meme theory of artifacts views them as agents competing with each other in an evolving environment. Memes are a type of life.

***Bifurcation***

Social movements, such as the stock market, are day-to-day unpredictable while being globally limited to some specific strange attractor. The most dramatic social upheavals occur when a fundamental change alters the system’s parameters, leading to a bifurcation and an entirely new computational strange attractor.

***On my McLuhanizing***

Never forget that Ph. D. stands for “piled high and deep.”

***Table of Emulations***

At this point, I’ve introduced all five of the computational levels that we’re going to think about. Now, in the next chapter (6) I’ll talk a lot about having computations of various kinds emulate each other. So at this point I’d like to look at some of the kinds of emulation that I have in mind.

| | PC | Fluttering leaf | Developing fetus | Human mind | The changing content of the web |
|---|---|---|---|---|---|
| PC | * | | | | |
| Fluttering Leaf | | * | | | |
| Developing fetus | | | * | | |
| Human mind | | | | * | |
| The changing content of the web | | | | | * |

**Table: Computations Emulating Each Other**

*The cells describe how a computation named at the top of the column would emulate the computation named at the left of the row.*

***Predicting a Baseball***

[Row deleted from the "Uses of Prediction" Table in section *6.3: The Need for Speed*.]

| | | |
|---|---|---|
| A high fly baseball. | Mental emulation of the trajectory. | You catch the ball. |

Actually, it seems that baseball players don't catch fly balls by emulating the trajectory so much as by running around and using a particular kind of feedback loop. This loop, called a linear optical trajectory, depends upon being in motion, which is why talented fielders tend always to make their catches on the run. See http://www.sciencenews.org/articles/20021012/mathtrek.asp.

***Dig at Theoretical Computer Scientists***

[Removed from the footnote on the hundreds of complexity classes discussed by theoretical computer scientists.] Not to ruffle any feathers, but my suspicion is that theoreticians distinguish so very many complexity classes because inventing a new definition is an easier way to generate a computer science dissertation or journal article than is designing, writing and debugging some interesting and useful code.

### *Self-mocking addendum to footnote on Complexity Classes*

Gabba-gabba, hey! (Said interjection signifying that I'm parroting words whose meaning is obscure to me.)

### *Limits of Philosophy*

Keep in mind that philosophical arguments tend not to reach any absolute conclusion. The real point of philosophizing is to open your mind, and to see the world more clearly. Even if we don't solve the question of whether "everything is a computation," when we're done discussing it, you may see things in a different and richer way than you did before.

### *Computations vs. Thoughts*

At this point we won't worry about the fact that there are practical limitations on what a given person or computer can actually do. Sure you can't memorize the Manhattan phone book, reorganize it the numerical order of the phone numbers — and then whittle the list down to the phone numbers that happen to be prime numbers (that is, numbers having no divisors other than one and themselves). But, given enough time (like a trillion years of one-day work weeks with ninety-nine day weekends), enough paper (like every tree on the planet), enough coffee and chocolate (well, maybe there isn't enough) — you could get the job done. And, putting the shoe on the other foot, your, laptop computer can't recognize the faces of your business acquaintances and prompt you with their first names as they walk up to you. But, given a video camera, and a particular kind of a program known as a "neural net," a few trillion bytes of RAM to be able to run a big neural net, a blazingly fast processor chip, and perhaps a thousand hours "training" the neural net on a large library of photos — your laptop could get the job done.

There are certain kinds of tasks that computers are hardwired to be good at, and certain kinds of tasks that human brains handle very easily. But with a little effort, and a little relaxing of realistic practical restrictions, we can imagine humans carrying out any computer task and we can imagine computers doing at least everyday algorithmic kinds of human tasks as well.

### *Wolfram's Science*

Although he studies a few continuous-valued computational processes, he prefers focusing on discrete-valued processes. The reason is that unlike real-number processes, no approximations are involved in discrete-valued computations. Wolfram feels that when you use real numbers there's a chance of the round-off errors dominating the phenomena.

In discussing possible applications of his work, Wolfram mentions the somewhat intriguing thought that nanotechnology could be based on small simple machines of the type that he likes to study.

### *Principle of Universal Unpredictability*

Wolfram's *Principle of Universal Unpredictability*: Most naturally occurring universal computations are unpredictable. This means that for most universal computations, there is no quick and dirty shortcut method for predicting the computation's result. This principle is supported both by Turing's theoretical work and by Wolfram's empirical studies of large classes of simply defined computations. And possibly by counting arguments: there are only so many short algorithms.

Combined with the second form of the PCE, the Principle of Universal Unpredictability says that almost all naturally occurring complex computations are unpredictable. This combination of the two conjectures merits its own name, the PCU, or Principle of Computational Unpredictability.

***Future History with Dates Left In***

| **Guessed Year** | **Future Technology** | **Description** |
|---|---|---|
| 2030 | Piezoplastic | Plastic whose colors and shape are dynamically controlled by electronic inputs. Usable as a non-boxy computer display. |
| 2040 | Lifeboxes | Artificially intelligent simulacra of people. |
| 2050 | Limpware Engineering | The science of programming piezoplastic. |
| 2060 | Dragonflies | Insect-sized flying camera eyes, individually owned (or rented) so people can see whatever they want. |
| 2070 | Sluggie processors | Personal computer processors are now made of soft plastic. |
| 2080 | Radiotelepathy | It becomes possible to electromagnetically send thoughts from brain to brain. |
| 2080 | The uvvy | The ultimate wireless device; the uvvy sits on your neck and gives you web, email, cell phone, and direct thought access. |
| 2090 | Recording dreams | A side-effect of the uvvy. Culture craze for dreams, society becomes surreal for awhile. |
| 2115 | Knife plants, House trees | Genetically engineered plants begin producing consumer goods, for instance knives. A largish specialized seed can grow you a house. |
| 2120 | Pet construction kit | People can program their own pet characteristics. Pet dinosaurs are very popular. |
| 2280 | "Aug dog" | It becomes popular to bioengineer your own body, these changes are called augmentations; thus the popular term for a body changers is "aug dog." |
| 2290 | Archipelago people | It becomes possible to have several disconnected hands or eyeballs that move about independently from your main body; you stay in touch using uvvies. |
| 2290 | Mermen , mermaids | Bioengineered people move into new niches like under the sea. |

| | | |
|---|---|---|
| 2350 | Programmable clones | It becomes possible to grow a clone of yourself in a tank and program its brain with the contents of your lifebox file, creating a person very much like yourself. |
| 3001 | The alla | The age of direct matter control arrives and we can change anything into anything. Matter is fully programmable. |
| 3002 | Space migration | People use allas to live in asteroids |
| 3003 | 3ox | A new technology for identically copying an existing objects, including living beings |
| 3150 | Ooies | Uvvies become internal organs, so that people are constantly in contact. Society truly becomes a hive. |
| 3400 | Colony people | Some individuals 3ox or clone hundreds of copies of themselves, with the copies connected via ooie. |
| 3400 | Spacebug people | Still more advanced bioengineering allows people to live in the hard vacuum of outer space. |
| 3666 | Teleportation | Insight into fundamental physics gives people the ability to jump to arbitrarily distant space locations. |
| 4050 | People free to move in higher dimensions | Travel to the other worlds beyond our space and time. |

### *Stepping Through an Argument for the Universal Unpredictability Lemma*

Let's consider the universally computing CA Rule 110 and see what happens if we try to carry out Wolfram's argument sketched on NKS, p. 742, that a universal computation is "computationally irreducible."

Specifically, let's look at a computation C110 such that if rowcount is an integer measuring the number of number of rows to be computed and if input is an integer that codes up a start pattern then C110(rowcount, input) = output, means that output is an integer coding up the rowcount$^{th}$ row of a Rule 110 computation starting with the pattern coded by input.

Normally, computing C110(rowcount, input) = output takes rowcount*input + $n^2$ individual cell-update steps or, if you prefer, rowcount row-updates.

Now suppose we have a prediction algorithm called Predictor such that Predictor(rowcount, input) = output maps rowcount-input pairs into output rows just like C110 does. And suppose that Predictor computes this in some very small number of computational steps PredictorRuntime(rowcount, input). We might imagine that for sufficiently large rowcount and input, this runtime number might be on the order of the logarithms of the second two arguments, that is, we might expect that

PredictorRuntime(rowcount, input) ≈ log(rowcount)+log(input)

Wolfram's argument suggests that we now use the universality of Rule 110 to let it simulate the computation Predictor(rowcount, input). Simulating Predictor means that we have a string prog coding up the program of Predictor and that we can combine prog with a desired rowcount count and an arbitrary input to make a new argument called, say, prog-rowcount-input. We feed prog-rowcount-input into C110, and by computing some as yet unknown number of rows simulationrowcount, the computation of C110(simulationrowcount, prog-rowcount-input) will emulate Predictor(rowcount, input) by generating output, that is by producing the rowcount$^{th}$ row of Rule 110 starting with input.

The interesting question is whether simulationrowcount can be less than rowcount.

If we are pessimists, then we might worry that the cost of having Rule 110 (and thus the computation C110) simulate a step of Predictor might be in fact arbitrarily large, subject to the current state of the computation in progress, so that simulationrowcount is very much larger than the modest size of PredictorRuntime(rowcount, input) would lead us to expect. If for instance we need to keep sending information back and forth through the computational space by means of "gliders," then it could be that, the larger the computation gets, the slower it runs — this would be akin to having to take into account the limitations of the speed of light in the functioning of a very large electronic computer.

But let's give Wolfram a break here, let's optimistically suppose that the way that Rule 110 emulates the computation of Predictor involves a fixed maximum cost of StepCost steps per computational step of Predictor. So then we would expect to have

simulationrowcount = StepCost * PredictorRuntime(rowcount, input)
≈ StepCost *(log(rowcount)+log(input))

The interesting situation of Rule 110 "outrunning" itself occurs if

simulationrowcount < rowcount

Wolfram seems to think that this situation would lead to a contradiction involving an infinite descending sequence of positive natural numbers derived from nesting the simulations.

But if you analyze it closely, the descending chain will stop as soon as one crosses a cost-benefit condition whereby the overhead of simulation cost swamps the gain of successive levels of simulation. [There will be two overhead costs to take into account: (a) the constant StepCost multiplier already mentioned and (b) the cost of inputting larger and larger strings as the initial conditions so as to simulate more and more refined kinds of computations.]

So my impression is that Wolfram's briefly sketched argument doesn't work.

### *The Universal Unpredictability Lemma*

I'm interested in finding some support for the following statement.

*Universal Unpredictability Lemma.*
Most naturally occurring universal computations are unpredictable.

As I mentioned before, by "unpredictable" I mean "not being emulable by a computation whose runtime is logarithmically faster." Wolfram prefers the word *irreducible* for what I call *unpredictable*.

Perhaps Turing's Corollary that universal computations are runtime unbounded might be viewed as a kind of inspiration for what Wolfram would really like to be true: that a universal computation must be unpredictable. But the two properties have no direct connection that I can see.

Wolfram offers the following very brief proof sketch of why a universal computation should be irreducible or unpredictable in the sense that there won't be a computation that consistently predicts the universal system's results faster than the system in question. [Wolfram omits to mention that he really means logarithmically faster, but he should, as it's completely trivial get linear speedups of any computation simply by using more states.]

"Consider trying to outrun the evolution of a universal system. Since such a system can emulate any system, it can in particular emulate any system that is trying to outrun it. And from this it follows that nothing can systematically outrun the universal system. For any system that could would in effect also have to be able to outrun itself." *A New Kind of Science*, p. 742.

Wolfram's idea seems to be that if we (*thesis*) have a predictor computation that emulates our universal computation with faster runtime than that of actually running the universal computation, then (*antithesis*) having the universal computation emulate the predictor should also be faster than the universal computation, and then (*synthesis*) having the predictor emulate the universal computation's emulation of the predictor should be faster yet, and now we can use the synthesis as a new thesis, and get a yet faster computation, and so on forever. But it would be impossible to have an endless sequence of smaller and smaller run times. Therefore a universal computation like can't be predicted.

But if you think through some specific cases, we can see that this particular argument doesn't work. The descending chain will stop as soon as one crosses a cost-benefit condition whereby the overhead of simulation cost swamps the gain of the successive levels of simulation. Put differently, the antithetic step won't work very often.

Let me refer back to the U and DumbU example in section 6.3 of the main text.

If we try and apply Wolfram's argument, we see that DumbU can emulate U. But, since DumbU is wired to carry out the wasteful tally doubling operation between each step, DumbU-emulating-U will run slower than DumbU, so the very first antithetic step fails and we never even take the first step of Wolfram's proposed infinite regress.

Falling back, we now ask if there is any kind of counting argument that might suggest that unpredictable universal computations are more common than predictable ones?

One thinks of Chaitin's argument that we can find arbitrarily long strings of zeroes and ones are random in the sense of not having a description shorter than themselves. To be more precise, Chaitin supposes that we fix a particular universal Turing machine U and that we say that string NameK is a *name* for string K if U with input NameK produces the string K and halts. For any string length n, there are $2^n$ strings of length n, but only $2^n - 1$ strings of length less than n, so at least one string of length n has no name shorter than n. And if we say string K of length n is predictable if it has a name of length at most log(n), then very many strings are unpredictable.

But this style of argument doesn't apply to universal Turing machines because when I say U is unpredictable, I'm saying there is no Turing machine V *of any size* such that V emulates U with a runtime eventually on the order of the log of U's runtime. Unlike in the Chaitin argument, we're not limited to looking at predictor Turing machines of size smaller than the object to be predicted.

Nevertheless, with Wolfram, I too, want to believe the Universal Unpredictability Lemma to be applicable to all of the interesting cases that I've discussed — including the dynamics of physical systems, the processes of the human mind, and the workings of human society.

### *Definition of Oracle*

- *Definition*. The computation *OP is an oracle for the computation P* iff
  (i) OP takes the same inputs as P, is everywhere defined, and has a special extra output state △.
  (i) If P(a) halts at b, then OP(a) halts at b.

(ii) If P(a) doesn't halt, then OP(a) halts at △.

And we say that *P has an oracle* iff there is some OP which is an oracle for P.

I won't use oracles for anything in this appendix, but as the notion of oracle seems more intuitive than the notion of a solvable halting problem, I use it in the main text. Before moving on, I'll show that the two notions are really equivalent.

*Proposition*. P has an oracle iff P has a solvable halting problem.

To see the truth of this proposition in the forward direction, note that if P has an oracle OP, you can use OP as a black box inside a new computation HP such that HP(a) outputs 0 as soon as OP(a) halts at △, and HP(a) outputs 1 as soon as OP(a) halts at any other value. To see the truth in the reverse direction, you can cobble together an OP by putting both HP and P inside a black box. Given an input a, you first feed it to HP. If HP(a) = 0, let OP(a) return △. If HP(a) = 1, then run the computation P(a) until it halts at some b and let OP(a) = b.

### ***Drafts For the Ending***

[Aug 25, 2004. I was going to call this last section "Wake Up".]

Although it may not seem like it, one of my goals in writing this book has been to liberate myself from computers. So why have I been writing about them at such length? I wasn't quite able to formulate my logic until I found the following remark by Marshall McLuhan. Though most of us imagine McLuhan to be a cheerleader for progress, the opposite was the case.

"I am resolutely opposed to all innovation, all change, but I am determined to understand what's happening. Because I don't choose just to sit and let the juggernaut roll over me. Many people seem to think that if you talk about something recent, you're in favor of it. The exact opposite is true in my case. Anything I talk about is almost certainly something I'm resolutely against. And it seems to me the best way to oppose it is to understand it. And then you know where to turn off the buttons."

In his later life, McLuhan rephrased his adage "the medium is the message" by saying that the best way to understand the effects of a new technology is to look not at the figure, but at the ground. Thus, rather than talking about what people use computers for, we might look at how they change people's behavior.

Think of the time you spend upgrading the software on your machine. Think of the toll the hours at the screen take on your wrists and your back. Go into a coffee shop and look at the people isolated behind their laptop screens. Walk down a street and see blank-faced people pecking at their wireless gizmos. Imagine a world where your time was your own.

When I say that everything in the world can be viewed as a computation, I'm not saying that PCs are as good as reality. Far from it. Yes, universal automatism teaches us that there's a common ground by which to compare nature to PCs. But on this common ground, we can readily see that the natural world is incalculably more powerful and interesting than the odd flickering boxes we're wedded to in the era Y2K.

The air is a gnarly ocean; the leaves dance on the trees.

Have pity on your tired eyes and aching back. Do something nice with your body.

*Who am I to tell you what to do? Well, actually this advice is for me.*

And look into your head. Underneath the planning and resenting and wanting and worrying is the river of thought. Look at it like you'd watch the ripples in a stream. It's beautiful. Let go of plans and expectations.

*All right!*

Other people are the most interesting and beautiful entities you'll ever see. Honor them, talk to them, accept them, love them.

*I'm sorry I've been hung up on this book so long, Sylvia. Let's play.*

This moment is the only moment you have.

*I'm turning off the machine and going camping. Good bye!*

[ And then I did go to the beach, and thought better of this ending and wrote a new one on August 28, 2004, with some advice and this closing.]

Here I am preaching platitudes at you. How pompous, how fatuous, how annoying, how *old*. I sound like my father.

Well, let's say the advice is actually aimed at me. I need it. I forget the simplest things.

And yes, I realize it's a stretch to say that these slogans follow from my book's long class four chain of reasoning. But they're a nice place to end up.

So now we're done.

And thanks for riding along. It's been fun.

[And then I saw this line in a book review of David Mitchell's *Cloud Atlas* in the New York Times: "Self-mockery as self-protection is a very old gambit, certainly but it is beneath a writer as brilliant a Mitchell." And I thought, whoah, I don't want to use self-mockery if it's beneath a brilliant writer!]

### *Caption to Post Problem Illo*

*If the PCE were true, the region labeled "Universal" could be expanded to include almost everything except the contents of the "Solvable" region. But this is impossible, as we know there are very many unsolvable non-universal computations.*

### *Remark on Gerry Sacks*

As a historical aside, let me remark that Gerald Sacks is an utterly charming man, and the only really sharp-dressed mathematician I've ever known.

### *Recursiveness as "Having a Solvable Halting Problem"*

We sometimes use the name recursive for a computation that has a solvable decision problem.

*Definition*. The computation P is said to be *recursive* iff it has a solvable halting problem.

The reasons for the name "recursive" are historical. In the 1930s, logicians such as Alonzo Church, Kurt Gödel, Jacques Herbrand, and Stephen Kleene — as well as Allan Turing — were investigating various ways to define elementary computations. One class of such computations was called the *general recursive functions*. General recursive functions map integers into integers, but they can have the property of failing to return any output at all for certain inputs. This is because the general recursive functions are allowed to use unbounded searches in their definitions — and sometimes these searches fail. When some particular input sets off an unsuccessful endless search, that means there's never going to be an output.

In the case where a general recursive function does happen to return an output for each input, we call it a *total recursive function*. When I say a computation is recursive in the sense of having a solvable halting problem, we really mean that it resembles a total recursive function — in the sense that, by combining P and its halt detector HP, we can create a computation P* which halts with an output for every input. P* acts the same as P on P's halting inputs, and it returns False for P's non-halting inputs. P* uses HP to decide which case to follow.

### *The Unsolvable Production Problem*

Suppose I have a computation P and I'd like to find a way, given an In and an Out, to find out if P(In) ever produces Out. As with the halting question, we can't distinguish the two possibilities simply by running P(In) and waiting for Out to appear.

*P(In) produces Out*. In this case, we'll be successful, and we know after a finite time that, yes, Out appears among the states produced by P(In).

*P(In) doesn't produce Out*. In this case, running P(In) never leads to an answer, because we'll never be sure if we've waited long enough for Out to appear.

The cases where P(In) doesn't produce Out lead to an unsuccessful search. We'd like to find a way to short circuit the endless wait for Out to turn up among the states produced by P(In). That is, we'd like to have an *non-production detector* computation PFailsToProduce.

*Definition*. Given a computation P, we say the computation PFailsToProduce is *a non-production detector* if PFailsToProduce has a special state True, and the following two conditions are equivalent

PFailsToProduce (In, Out) produces True
P(In) doesn't produce Out.

*Definition*. A computation P has a *solvable production problem* iff it has a non-production detector. Otherwise we say P has an *unsolvable production problem*.

*Turing's Theorem (Variation 2)*. If U is a universal computation, then U has an unsolvable production problem.

### *Degrees of Unsolvability*

It's convenient to use the notion of emulation to compare the power of computations.

*Definition*. Let P and Q be computations. If Q can emulate P, we say *P has an emulation degree less than or equal to Q*, and write $P \leq_e Q$.

If Q can emulate P and P can emulate Q, we say *P has the same emulation degree as Q*, and we write $P =_e Q$.

If Q can emulate P and P can't emulate Q, we say *P has a smaller emulation degree than Q*, and we write $P <_e Q$.

It's not hard to prove that having the same degree is a transitive relationship; that is,

If $P =_e Q$ and $Q =_e R$, then $P =_e R$.

This means we can use $=_e$ to divide the set of all possible computations into equivalence classes of computations such that all the computations in a given class can emulate each other.

One can also prove that $=_e$ preserves the property of having a solvable halting problem. For this reason we can speak of the $=_e$ equivalence classes as *degrees of unsolvability*. And we can extend the relations $\leq_e$, $=_e$, and $<_e$ to these classes in an obvious way.

Now suppose that R is some very simple computation which is everywhere defined — to be specific, suppose that, given any input In, the computation R(In) simply stays in the In state forever. Any computation at all can emulate R, but we don't expect that the do-nothing R can emulate all the other computations. In symbols, if P is any other computation, R $\leq_e$ P and possibly R $<_e$ P. For this reason, we say that R represents a minimal degree of unsolvability. Abstracting a bit, we use the symbol $\mathcal{R}$ to stand for R's degree of unsolvability.

Do note that, due to the fact that the translations used in the definition of emulation can be any everywhere defined computations at all, it turns that that *any* everywhere defined computation will be found in $\mathcal{R}$. Less obviously, any computation that has a solvable halting problem relative to the default target detector will be in $\mathcal{R}$ as well. So the degree $\mathcal{R}$ represents a fairly large class of computations. $\mathcal{R}$ is sometimes known as the recursive degree.

Now suppose that U is a universal computation. Since U can emulate any computation at all, but some computations, such as R will not be able to emulate U. That is, if P is any other computation, P $\leq_e$ U and possibly P $<_e$ U. In other words, U represents a maximal degree of unsolvability. Abstracting as before, we use the symbol $\mathcal{U}$ to stand for U's degree of unsolvability. Given that $\mathcal{U}$ contains all of the various universal computations, $\mathcal{U}$ is also a large class of computations. $\mathcal{U}$ is sometimes known as the universal degree.

Because the three facts that (by Turing's' theorem) U has an unsolvable halting problem, that (since its everywhere defined) R has a solvable halting problem, and that (as can be proved from the definition of emulation) $=_e$ preserves the property of having a solvable halting problem, we can deduce the following proposition.

*Proposition*. There are at least two distinct degrees of unsolvability, the recursive degree and the universal degree. In symbols, $\mathcal{R} <_e \mathcal{U}$.

So now we know that we have at least two degrees of computations: the minimal degree $\mathcal{R}$ of the computations that are everywhere definable or which have solvable halting problems, and the maximal degree $\mathcal{U}$ of the universal computations.

In 1940, the logician Emil Post posed *Post's Problem*, asking if there any intermediate degrees. We can formulate Post's Problem in various ways.

Are there any intermediate unsolvability degrees $\mathcal{M}$ such that $\mathcal{R} <_e \mathcal{M}$ and $\mathcal{M} <_e \mathcal{U}$?

Is there a computation M such that M has an unsolvable halting problem, but M is not universal?

Why is the second formulation is the same as the first? Assuming $\mathcal{M}$ is the degree of M, the condition $\mathcal{R} <_e \mathcal{M}$ is equivalent to saying that M has an unsolvable halting problem.

And $\mathcal{M} <_e \mathcal{U}$ is equivalent to saying that M isn't universal.

Post's problem was solved in the affirmative by Richard Friedberg and Albert Muchnik, working independently in 1956. And further work by mathematical logicians such as Gerald Sacks has shown that the degrees of unsolvability represent about as messy and unruly an ordering as one can imagine.[10]

There are *infinitely* many distinct degrees of unsovability between the recursive and the universal degrees.

The degrees are *dense* in the sense that if $P <_e Q$, there is an S such that $P <_e S$ and $P <_e S$.

And to make things gnarlier, the degrees of unsolvability *don't fall into a linear ordering*, that is, we can find P and Q such neither $P \leq_e Q$ nor $Q \leq_e P$ is true, that is, P can't emulate Q, and Q can't emulate P,

### *Historical Analogy Regarding Intermediate Degrees*

*History*. One might say that the Greeks worked primarily with real numbers that can be expressed either as the fraction of two whole numbers, or which can be obtained by the process of taking square roots. By the time of the Renaissance, mathematicians had learned to work with roots of all kinds, that is, with the full class of algebraic numbers — where an algebraic number can be expressed as the solution to some polynomial algebraic equation formulated in terms of whole numbers. The non-algebraic numbers were dubbed the transcendental numbers. And, for a time, nobody was sure if any transcendental numbers existed.

*Analogy*. Until the mid-1950s, it seemed possible that, whatever precise notion of degrees of unsolvability one uses, there might be only two degrees of unsolvability among the computations: the recursive degree $\mathcal{R}$ and the universal degree $\mathcal{U}$. Nobody was sure if intermediate degrees existed, or if there were non-universal computations that had unsolvable halting problems.

*History*. The first constructions of transcendental real numbers were carried out by Joseph Liouville, starting in 1884. Liouville's numbers were, however, quite artificial, such as the so-called Liouvillian number 0.1100010000000000000000010000... which has a 1 in

---

[10] See the books on recursion theory referenced at the beginning of our Technical Appendix, and see Richard Shore, "Conjectures and Questions from Gerald Sack's *Degrees of Unsolvability*", *Archive for Mathematical Logic* **36** (1997), 233-253. The paper is available online at, http://www.math.cornell.edu/~shore/papers/pdf/sackstp3.pdf.

In point of fact, the Friedberg, Muchnik and Sacks results regarding degrees of unsolvability were proved for an ordering $\leq_T$ known as Turing reducibility rather than the emulation degree comparison $\leq_e$ that I'm using here. My $\leq_e$ is in fact equivalent to what recursion theorists call one-one reducibility $\leq_1$. But for $\leq_1$ it's also the case that pairs of incomparable degrees exist, that the degrees are dense, and that there are infinitely many degrees. This follows as a corollary to the Friedberg, Muchnik and Sacks results for $\leq_T$ because $\leq_1$ is a weaker notion than $\leq_T$. And independent proofs of some of the same facts about $\leq_1$ result from work by J. C. E. Dekker, also in the 1950s. Dekker was my professor in a class on Recursion Theory at Rutgers, years ago. I used to think I had an Erdös number thanks to Dekker, who wrote a paper with my thesis advisor Erik Ellentuck, whom I also wrote a paper with, but I was mistaken, Dekker didn't write a paper with Paul Erdös, so far as I know. But Ellentuck also collaborated with Richard T. Bumby on a paper, whose Erdos number is 2, by way of R. Silverman who wrote papers with both Bumby and Erdos, so my Erdos number is (at most) 4. Thanks to Jerry Grossman, http://personalwebs.oakland.edu/~grossman/, for explaing this to me.

the decimal positions n! and 0 in all the other places. Someone might readily say that a number like this is unlikely to occur in any real context.

*Analogy*. In 1956, Richard Friedberg and Albert Muchnik carried out a complex and artificial construction of a computation of intermediate degree for one notion of solvability degrees and logicians such as J. C. E. Dekker did similar constructions for other notions of degrees.

*History*. In 1874, Georg Cantor constructed a specific enumeration of the countable set of algebraic reals and produced a transcendental number by having it differ in the $i^{th}$ digit from the $i^{th}$ member of his enumeration of the algebraic numbers. Again, someone could say that Cantor's number isn't a number that would naturally occur, that it is artificial, and that it depends in an essential way upon higher-order concepts such as treating an infinite enumeration of reals as a completed object.

*Analogy*. In the 1960s, the logician Gerald Sacks and his colleagues streamlined the techniques of Friedberg and Muchnik to produce a bewildering variety of computations of intermediate degree. Yet all of their constructions remained somewhat artificial.

*History*. But in 1873 Charles Hermite proved that the relatively non-artificial number e is transcendental, and in 1882 Ferdinand Lindemann proved that the downright garden-variety number pi is transcendental as well.

*Analogy*. Hasn't happened yet — and may never happen. But my guess is that eventually someone will prove that a natural and familiar computation like CA Rule 30 is a non-universal computation that has an unsolvable halting problem.

### *Weinberg's Zinger*

And of course Turing machines and two-state CAs are, for instance, much slower than a PC. As the physicist Stephen Weinberg remarks, "This is why Dell and Compaq don't sell Turing machines or rule 110 cellular automata." Stephen Weinberg, "Is the Universe a Computer," *The New York Review of Books,* October 24, 2002, pp. 43- 47. This was one of the many smug and negative reviews which greeted the publication of Wolfram's *A New Kind of Science*. As John Walker remarked about the reviews, "Most of them were An Old Kind of Envy." Something I found particularly wrong-headed in Weinberg's review was his cavalier remark that we don't really want to know the outcomes of complex processes like medium energy particle collisions. But I have to admit Weinberg got off some good zingers, as when he attributes Wolfram's universal automatism to excessive time spent in programming, and compares his world-view to that of a carpenter who looks at the moon and wonders if it might be made of wood.

### *Joke About Cosmic Fry's Electronics*

If you do find your way to that Fry's in the Sky, get me a slow-down ray! (Ignore this, I'm slap-happy, punch-drunk, on the ropes. I've been thinking too hard about this stuff for way too long.)

### *Everyone's Different*

It's no accident that every single person in the world looks different. Even if two fertilized eggs were to have the same DNA, the biochemical differences between the cells' cytoplasm would have an effect, as would the differing environments of their mothers' wombs. Even identical twins are different, if you take a magnifying glass to them.

### *Stimulants*

The stimulant issue is a scam, an irrelevant sideshow. A trick for suckers. Who ever got smarter by taking a pill?

### *Television*

When I watch television, I feel that I really *am* seeing predictable computations. News, commercials, entertainment — it's all so stereotyped, so lifeless, so utterly false. Nobody in real life acts like the people on TV, not even like the people on *reality* TV, for the reality TV shows have been carefully edited to remove any trace of gnarly originality. If something unpredictable every happens on TV, the U. S. Congress angrily decries the anomaly for months.

Whoah there, querulous geezer, there's more to society than TV.

### *Quantum Computation and Predictability*

You know how sometimes you're at a restaurant with five or six other people, and at the end of the meal everyone's paying their share of the check, tossing hard-earned cash dollar bills down on the table. And then there's always the one *schmuck* who loses control at the sight of the cash and grabs it all up and says they'll pay with their credit card (so they can get air miles, put the meal on their expense account, avoid an auto teller fee, and come out ahead by pocketing most of the tip).

To my way of thinking that's what quantum computation is like, sitting at the table of computation theory, scooping up the hard cash of informed speculation, and claiming it can render any process predictable — if you'll just trust its credit card.

### *Gödel's Second Incompleteness Theorem*

We can formalize the proof of Gödel's First Incompleteness Theorem within the formal system F itself. To do this, we represent the sentence "F is consistent" by a sentence Con(F) of the form " '0=1' is not a theorem of F." By a mind-breaking feat of jumping out of the system, Gödel showed how one can in turn carry out this proof within the formal system F itself, to establish as a theorem a statement of the form "if Con(F) then $G_f$" As a consequence, Gödel draws a Second Incompleteness Theorem.

*Gödel's Second Incompleteness Theorem*. If F is a consistent finitely given formal system as powerful as arithmetic, then the sentence Con(F) is undecidable for F.

### *Chaitin's Proof*

In order to describe Chaitin's proof, I need another definition relating to Turing machines that use the two symbols 0 and 1. Recall that in the Chapter One we discussed the notion of adopting a fixed enumeration of these Turing machines so that for an integer e, $T_e$ is a Turing machine. We can view $T_e$ as a kind of name for a string n by writing $T_e(0) = n$ to mean that the computation $T_e(0)$ halts and that when $T_e(0)$ halts, the binary expression for the string n is found on the Turing machine tape.

*Definition*. The integer n is Chaitin-Kolmogorov *incompressible* if for no e < n do we have $T_e(0) = n$.

Rather than working with arbitrary kinds of computations, the proof of Chaitin's theorem focuses on Turing machines with two symbols. The key notion behind the proof is that both the programs and the outputs of these Turing machines can both be thought of as strings of 0s and 1s. Because of this, a program can itself be regarded as possibly being the output of a smaller program. For any P, let P* be the program such that if P(0) halts with output Q, Q is treated as a program and Q(0) is run. There will be some fixed number overhead such that size(P*) = size(P) + overhead.

Now Chaitin argues that if of, once again, the unsovability of the halting problem. know that a string is a name for another string, you need to know that, when fed into the universal computer, the name produces a computation that *writes the target string and halts*. So proving that a number is Chaitin complex involves proving that certain computations don't halt and therefore fail to name the string. And, as we know by now, some computations never halt even though you can't prove this to be so.

A more transparent way to get at the proof of Chaitin's theorem is to use the following line of thought. If that if Chaitin's theorem weren't true, then T could prove there is a number called BerryT which is definable as "the first number that doesn't have a definition shorter than itself." And then T would be in trouble as this proof provides a short name for BerryT.

### *People Who Email Me About Gödel's Proof*

I've written about Gödel's proof before, like in *Infinity and the Mind* and in *Mind Tools*, and I get email about it every few months, usually from people who think the proof is wrong. Guys wanting to tell me they're smarter than Kurt Gödel. "*Oooooh* kay."

### *Adding Quantifiers*

The simple sentences G I've been discussing can be characterized as having a single kind of quantifier — where a quantifier is a phrase like "for all n" or "for some n". Although we've speaking of G as having the form "For all n, g[n] is false," we could also express it as the negation of a sentence of the form "There is an n such that g[n]."

Adding more quantifiers of the same kind doesn't change much. But if we allow alternating quantifiers we get into a richer zone of undecidability.

### *Course-of-values Simulation*

A slightly stronger form of simulation is what one might call course-of-values simulation. Say that Big with translation function tr is a *course-of-values-simulation* of Small iff, for any s0, if the computation Small(s0) produces, at some increasing sequence of times, the output states s1, s2, s3, etc., then the Big(tr(s0)) will also produce, for some increasing sequence of times, the states tr(s1), tr(s2), tr(s3), etc.

## *References*

David Deutsch, *The Fabric of Reality: The Science of Parallel Universes — and Its*

*Implications* (Penguin Books, 1997). Great, but way flaky.

## Books To Look At

Steven Strogatz, *Synch*. Has 3D Zhabotinsky rules.
Steven Johnson, *Emergence*. Seems important.
Matthew Derby. *Super Flat Times*. SF stories as mainstream.
Cohen and Stewart, *The Collapse of Chaos*.
Emanuel Rosen, *The Anatomy of Buzz*. Hubs.

## *Notes on* Linked

by Albert-László Barabási (Plume, New York 2003)

Considerable bragging and self-aggrandizing over essentially one single non-obvious result, viz. that building a network by adding nodes with each new node linking to existing nodes with probability based on the existing nodes' links leads to a power-law network with exponent -3.

(1) Intro: preliminary strutting and wild promises

(2) Erdös studied random networks generated by taking a set of N nodes, listing all pairs of nodes, and using a randomizer to connect each pair with some fixed probability p. A cluster is a set S of nodes so that for ever a and b in S, there is a series of links leading from a to b. Erdös and Rényi discovered a "percolation" effect, or phase transition, whereby if each node has, on the average, at least one link, there will be a cluster including most of the nodes. That is, as the average number of links per node increases, the number of nodes left out of the giant cluster decreases exponentially.

(3) Consider the network whose nodes are people and whose links are the "knows personally" relationship. Empirically, it seems to take less than six links to get from any node to another in P. Or the network W with pages as nodes and hyperlinks as links. Usually takes less than 19 links to get from one page to another.

(4) Say a node n is linked to k nodes. These neighbor nodes have a maximum possibility of (k choose 2) links among them. If the actual number of links among the neighbor nodes is m, we say that the clustering coefficient at node n is m/(k choose 2).

Watts and Strogatz observed that if you have a model with relatively high clustering, but with some constant number of relatively few links at each node, then adding a few long distance links decreases the average link distance between nodes. Duh!

(5) On the web, a few nodes have many more links than the others. Duh! These are called hubs.

(6) Vilfredo Pareto formulated the 80/20 rule. 80 percent of the peas come from 20 percent of the peapods. 80 percent of the publications come from 20 percent of the faculty. 80 percent of the links on the Web point to 15 percent of web pages.

A power law function relates a dependent variable y to an independent variable x in a certain way. Let u and y be the base ten logarithms of x and y, that is, suppose that $x = 10^u$ and $y = 10^v$. A typical power law has the form $y = 10^v = 10^{(b + alpha*u)} = 10^{(b + alpha*log(x))}$. If you plot it on log log paper, you get v = b + alpha*u, a line.

So really a power law arises simply when y varies as some power of x. Mandelbrot writes the most general form as $y = y_0(x + xshift)^{alpha}$ on p. 240 of his *Fractals* (1977). The xshift can be thought of as a shift along the x axis, and we normally prefer to simply work with x' = x + xshift.

Examples of alpha and b.

| x y Formula | log-log u v formula | alpha | b | xshift |
|---|---|---|---|---|
| y=1/x | v = -u | -1 | 0 | 0 |
| y = 1000/x | v = 3-x | -1 | 3 | 0 |
| $y = x^2$ | v = 2 * u | 2 | 0 | 0 |
| $y = 10^x$ | $v = 10^u$ | (need double log paper) | | 0 |
| y = K/square root(x) | v = log(K) - 0.5*u | -0.5 | log(K) | 0 |
| $y = y_0(x)^{alpha}$ | v = $\log(y_0)$+alpha*u | alpha | $\log(y_0)$ | xshift |

I wrote some examples of power laws on a piece of paper and lost them, so lets try again.

| x axis | y axis | Pairs... | | | | Formula |
|---|---|---|---|---|---|---|
| Size of advance | Number of writers getting this advance | $10^7$, 1 | $10^6$, 10 | $10^5$, $10^2$ | $10^2$, $10^5$ | v = 7-u; $y = 10^7/x$ |
| Number of links | Number of websites of this size | | | | | |
| Ordinal rank of word popularity | Probability that a word is this word | 1, 0.5 | 2, 0.5 | | | |

(7) The rich get richer algorithm can account for power law clustering, it comes out with an exponent of -3. Two key steps (a) growing network with new nodes arriving forever and (b) new nodes form links with existing nodes with probability proportional to the number of links the existing nodes have. I read his *Science* paper on this too. This is essentially the only result in his whole book.

(8) In some networks one node wins and they become stars.

(9) Scale free networks are vulnerable in that if the highly linked hub nodes are knocked out the network breaks into pieces.

(10) In a scale free network a virus can spread more easily because when the viruses get to hubs they radiate out a lot.

(11) Can the internet become self aware? Don't know.

(12) The Web actually breaks into pieces, given that links are directed. Central Core is mutually linked. In continent has links into Central Core, but you can't get back to it. OUT continent has links from Central Core but no links back to Central Core. Islands have mutual links but don't like back to core. Tubes run from IN to OUT.

(13) Can make a network for the many biomolecules in a cell, saying that two are linked if they participate in a reaction. Seems to be three degrees of separation typical in this small world network. Also seems to have a scale free topology.

(14) If nodes are fat cats and a link is being on the board of a corporation we get a scale free network.

(15) More bragging.
(16) Fit modularity into scale free networks. More bragging.

## *Journal*

### *March 26, 2003. Hiking in Big Sur, Waiting for Inspiration.*

I keep wondering what to do next, now that *Frek and the Elixir* is all but done. The rewards for my SF novels are so meager. I feel like giving up on SF. I can't face the tedium of writing a popular science book about computation, though. And I don't think there's a market for enlightenment books. If the reception of Bruegel had been less tepid, I'd say do my novel about Bosch's life. Right now any of these writing plans feels like beating a dead horse. Maybe polish and publish my journals? Write a third comic Silicon Valley novel, a bit less SF-ictional than before, possibly centering on Wolfram's *New Kind of Science*? But then I'd have to (ever more inaccurately) try and imagine young people. Maybe a non-book about computer science. Aphorisms. *Everything I Need to Know My Computer Taught Me*. Or forget about writing entirely. Paint. Take art classes and get better at it. Or get into computer programming again? *No way*. Maybe, yeah, paint and learn to paint like my man Jeroon van Aken a.k.a. Hieronymus Bosch.

### *March 27, 2003. The Idea!*

By the ocean, sun going down. On the rocky Esalen beach, alone, below the house where Terence McKenna and I led a seminar five or six years ago, maybe in 1997. I have an urge to write the *N. Y. Times Book Review* a letter defending Terence against a reviewer who, last week, said that, in Terence's last days, you couldn't tell if you were talking to him or to his brain tumor. Actually Terence was the same all the time. A seagull looks at me, its eyes disappear when seen directly head on. I sketch him in five or six positions: staring out to sea, cawing, looking at me, looking down at his feet, glancing at the shore. Sulfur smell from the stream raging into the sea. The sea here somehow wholly unlike in Santa Cruz. It's Big Sur sea, nay, *Esalen* sea. How lucky I am to be here. I say, "I love you," to the seagull. He bows. We do it again. Maybe the seagull is Terence.

Book idea: *Memoirs of a Crazy Mathematician*. Settling scores, taking credit. If I wrote a memoir, I wouldn't have to learn anything new, and I could talk about myself all the time. I'm old enough. Fifty-seven. That's really, really old. If the book did well, I could trundle out my collected journals. There might be new interest in the novels as well. Book as press-kit.

Patriotism is the last resort of a scoundrel. A memoir is the last resort of a writer.

### *April 16, 2003. What?*

Today I noticed a Memoir directory on my hard drive, and failed to recognize it. Memoir? Huh? Then I found this notes document again.

I'm leaning towards making the book less focused on being a memoir and more on the What is Everything idea.

Cleaning out the physical stack of folders on bookcase, I noticed those Web Mind folders, containing my essays purporting to explain the Web as a fractal in cyberspace or something like that.

***May 23, 2003. The Answers Spreadsheet.***

I worked this up for a guest lecture to Ralph Abraham's "Chaos and Complexity for Non-Scientists" class. All the "Answers" I've believed. I took out some of them for the book.

**"The Answers", by Rudy Rucker. 5/23/2003.**
**(Key: A=Art, B=Body & Senses, H=Human Society, M=Math, Science, Computers, P=Philosophy)**

| **Answer** | **Key** | **Everything Is...** | **In My Life** | **Year** | **My Age** |
|---|---|---|---|---|---|
| | | | | | |
| Family | H | Memories, talk, sharing | Childhood before differentiation. | 1946 | 0 |
| Society | H | Rules, strangers. The pecking order. | School. A heavily pecked 4th - 9th grade chicken. | 1954 | 8 |
| Self | P | Thoughts, reflections, desires, fears, plans, simulations, fantasies. | Autumn, dusk, I'm 9 years old, waiting for Mom to pick me up at school, a moment of feeling my future self looking back at this moment. | 1955 | 9 |
| Religion | P | God | Try to take church seriously, don't succeed. "You've got to be kidding." | 1958 | 12 |
| Science Fiction | A | Sense of wonder, goofs and eyeball kicks, transformation of mundane reality. | Random stories in anthologies. Heinlein, Sheckley, Dick, Lovecraft. Gibson, Sterling, Laidlaw and me (13 SF novels.). Cory Doctorow, Charlie Stross. | 1959 | 13 |
| Nature | B | Alive | Black Forest. Walking in the pastures near my house. The flow of the brook. | 1959 | 13 |
| Literature | A | Stories. Characters. Perceptions. Internal monologue. Personae to don. Creating a literary school. A sensibility. | Joyce. Kerouac. Burroughs. Pynchon. Borges. Poe. I've tried my hand at Lit, making my SF literary and writing a historical novel, also essays and memoirs. | 1964 | 18 |
| Science | M | Explanations via laws about unseen primitive entities: atoms, heat, magnetism, fitness. | Reading Scientific American, popular science books, didn't learn much in college, couldn't integrate it. | 1964 | 18 |
| Sex | B | Orgasm. Pure pleasure. Sensuality. Skin as sense organ. Stop thought. | Didn't really get into the deep aspects till after marriage. Sex takes practice. | 1966 | 20 |
| Marriage | B | Companionship, love, sharing, communication at a near telepathic level. | Somehow my wife and I have stayed together for 38 years now. There's a sense in which she's my best connection to my youth. She was there. | 1967 | 21 |
| Music | A | Beats, chords, embodied logic, emotional color. | Bo Diddley, Flatt & Scruggs, Beatles, Zappa, Stones, Ramones, NOFX, it goes on and on. | 1967 | 21 |
| Politics | H | Power | War. Fear, greed, hatred. A steady drain on psychic energy and mental equilibrium. | 1968 | 22 |
| Math | M | The bare forms of thought. Equations. | Grad school, teaching Calculus and finally understanding it. My survey *Mind Tools*. | 1968 | 22 |

| | | | | | |
|---|---|---|---|---|---|
| Children | B | Passing on information. DNA + nurture + teach + love. | I often think that the best thing I ever did (or luckiest thing that happened to me) was to have three children. | 1969 | 23 |
| Logic | M | Rules of deduction applied to axioms. | Studying mathematical logic. Meeting Kurt Gödel. | 1970 | 24 |
| Mysticism | P | One | The Perennial Philosophy, as Aldous Huxley called it. Seems obvious, uncluttered, true. | 1970 | 24 |
| Teaching | H | Passing on memes. Watered-down parenting. | Altruistic social good. Learn on the job. Good work hours. Performance art. | 1972 | 26 |
| Space | M | Higher dimensions. Curved space. Spacetime. Sheets of spacetime. | My first teaching job, I lectured on this. I wrote two books on 4D, the best known is *The Fourth Dimension*. | 1974 | 28 |
| Infinity | M | Infinite sets make up the forms of reality. We can, like, pixilize the world. | My thesis work was on Set Theory. I liked finding infinity in mathematics as this seemed almost like mysticism and theology. My *Infinity and the Mind*. | 1976 | 30 |
| Fitness | B | The body | Jogging, cross-country skiing, cycling, yoga, back-packing. Runner's high. Pleasure of the rushing motion. Wringing out the pain with yoga. | 1978 | 32 |
| Publishing | H | Teaching without face-to-face. | Constant struggle to get in print. Blogging lets everyone do it! | 1980 | 34 |
| Fractals | M | Infinitely detailed self-similar patterns. (Synthesize infinity and space). Gnarly. | The Mandelbrot set. A new paradise. And I needed the microscope of a computer to explore it. Fractal patterns seem to go hand in hand with chaotic motions. | 1984 | 38 |
| Cellular Automata | M | Very simple locally based rules that act in parallel, the same rule everywhere. Gnarly patterns and behaviors emerging. Gliders. | Life, Brian's Brain, Vote, Ranch. Visiting Toffoli and Wolfram in 1985 was a conversion experience. Writing CAs in assembly language. Using the CAM-6. Working with John Walker at Autodesk to make the *Rudy Rucker's CA Lab* package. Creating the CAPOW package with my students at SJSU. | 1985 | 39 |
| Artificial Life | A | Autonomous agents, don't need synchronization. Interacting. Simulations. Use genetic algorithms to evolve. Improve DNA-like genomes by fitness proportional reproduction. Fitness = gnarliness? | The exciting early Artificial Life conferences. My *Artificial Life Lab* package from the Waite Group. The simulated robotic evolution in my *Ware novels: information evolves away from the robot hardware to soft plastic moldie limpware to the freeware of intergalactic wave signals. | 1988 | 42 |
| Chaos | M | Deterministic yet unpredictable (in practice) processes. Wander around upon a characteristic attractor occasionally hopping to a new attractor. Gnarly things have fractal "strange" attractors. Why? | The 1980s computer scene in California. Key insight: Look at the motions of objects in nature, e.g. swaying branches. Looking at a tree combines Nature and Chaos, not to mention Self as you reflect on diggin' it. Creating the *James Gleick's Chaos the Software* package at Autodesk, (which includes fractals). | 1988 | 42 |
| Love | P | Opening my heart. | The universe loves itself. Yosemite vision. Love is practical in any situation: nothing else really works. | 1992 | 46 |

| | | | | | |
|---|---|---|---|---|---|
| Virtual Reality | A | A computer graphical simulation of reality. | The cyberspace craze. *Mondo 2000* viewing it as an immaterial new drug. | 1993 | 47 |
| Serenity. | P | Be kind, if only for selfish reasons: it helps you stay serene. | I'd always wanted to find Enlightenment and it had never once crossed my mind that the Quest might have something to do with trying to be a better person. | 1996 | 50 |
| The Web | H | Emergent global mind. The Library of Babel. | The Web isn't like *ugh* television because you can do-it-yourself. | 1998 | 52 |
| Painting | A | Blending colors and forms. An image. An idea made solid. | Learning to paint so I could write *As Above, So Below: A Novel of Peter Bruegel*. | 1999 | 53 |
| Software Engineerin g | A | Patterns. Classes interacting with each other. Like logic, but a living logic. | A slippery subject to teach. Inherently so? Jon Pearce teaching me about software patterns. | 2000 | 54 |
| Coherent mixed state conscious-ness | P | Merging with reality. To be "coherent" is to be a pre-collapse state of mind, to not have specific opinions. To adopt one position or another is to be decoherent. Wave with it. | My Leuven lectures, Fall, 2003, Satori in Paris. AI shows that any mental process we can explicitly describe can be simulated by a computer. But we "know" we are more than a computer program. The missing ingredient is of necessity not logically describable. Nick Herbert's "Quantum Tantra" says to view it as pre-wave-function-collapse merging. www.southerncrossreview.org/16/herbert .essay.htm | 2001 | 55 |
| Computati on | M | CAs, fractals, chaos, software engineering, virtual reality, artificial life all rolled into one. | Wolfram's *A New Kind of Science*: the dream of an incredibly simple rule that generates everything. | 2002 | 56 |
| Computer Games | A | Bring it all together on a computer: Graphics, AI, artificial life, chaos, story, art, sound. | Pac-Man to now. I teach a course on game design for five of ten years. I write a book *Software Engineering and Computer Games*. | 2003 | 57 |
| Pluralism | P | All the answers at once. Why should the world be simple, anyway? | William James's book, *A Pluralistic Universe*. He brings out the point that Monism could well be wrong. | 2004 | 58 |

I'm still thinking about *The Answers*, about all the various things I've thought, over the years, to be "the answer." (curved space, infinity, fractals, chaos, alife, CAs, nature, God, drugs, sex, alcohol, science fiction, literature, art, music, bicycling, consciousness, quantum mechanics, pluralism...) Wolfram's *New Kind of Science* would be a chapter of this book, but certainly not the whole book.

Problem is, I'm not sure how to structure *The Answers*. The most obvious form would be a memoir, my life in science, though I question if I'm of sufficient stature for Joe or Betty Shopper to care about my life in science. What would be cooler would be a series of Italo Calvino or Borges-like minifictions.

In particular I would prefer to avoid having this get into the "beating a dead horse" territory, that is, avoid simply repeating thoughts I already expressed in my other non-fiction books. Of course lots of writers do repeat their books, and with some success, so maybe this wouldn't matter.

I'm a little worried that writing about computers bores me? Well, it is what I know, and what is, in some sense, expected of me by now. And I really do have kind of a zest for it.

Thing is, I need an angle. Like Hofstadter obsessed on self-reference and he came up with *Gödel, Escher, Bach*.

Really it might be best to make a clean fresh start on a book, though, rather than dragging in the older material.

My main experiential teaching relates to how learning all these different things has influence how I perceive the world: nature, my mind, other people, society, etc.

There's a highly relevant bit in Kushner's *Masters of Doom*, about John Carmack, about how the years of graphics hacking have given Carmack a better appreciation of the world.

"In the shower, he would see a few bars of light on the wall and think, Hey, that's a diffuse specular reflection from the overhead lights reflected off the faucet. Rather than detaching him from the natural world, this viewpoint only made him appreciate it more deeply. 'These are things I find enchanting and miraculous,' he said. 'I don't have to be at the Grand Canyon to appreciate the way the world works, I can see that in reflections of light in my bathroom." p. 295

### *June 7, 2003. Book on Computer Games?*

I could worm my way into Electronic Arts via my ex-student Al Borecky, and into Valve Software via my friend Marc Laidlaw. This is catch-up ball, though it's for something I find fairly interesting. It could also be a "stations of the cross" book (my term for a non-fiction book where the author goes around interviewing luminaries in a field — there have been pukefully many books like this about computers and complexity, but not so many about games).

Maybe I could do a *Soul of a New Machine* or *Hacker Crackdown* kind of book about the computer game industry, which I seem to be backing into in various ways (teaching a course, writing a textbook and framework, going to conferences, visiting students and friends at their game companies, and finally buying a PlayStation 2) . I could maybe get a big enough advance to take a semester off from teaching and do lots of leg work ("stations of the cross.") I'd really be interested in anything you or the Media Lab is doing on the game front by the way. I do think it's only a matter of time now till we see lots of CAs in games.

Games have (1) Physics Simulation, (2) Graphics Hardware, (3) 3D graphics algorithms, (4) Art: Meshes, bitmaps, skins, texture, (5) Level designs, (6) AI (7) Sound, (8) User Interface, (9) Story, (10) Web connectivity, (11) Product placements.

But now I got three books on computer games from Amazon and am rethinking the reality of this. I read David Kushner, *Masters of Doom* in two days, it's about John Carmack the hacker and John Romero the gamer, creators of Doom and Quake. A great book, reminiscent of Stephen Levy's *Hackers*. I wonder how well the book is doing. Like how big is the market for that kind of book. Kushner bagged great blurbs from A-list pop nonfiction writers. He says the book took him six years and hundreds of interviews. I can't visualize doing that kind of project at all. And it wouldn't even be about me, me, *me*!

One phrase I saw in the book and liked a lot was that in Doom there's a type of enemy called a "Former Human."

### *June 9, 2003. The Quantum Mind.*

The nature of consciousness. I see this more as *part* of a book than as a *whole* book, though. The idea is base a line of thought on my Leuven lecture notes, *On Computation* and on "The Missing Mind," the paper-in-progress I'm ever-so-slowly writing with my Leuven philosopher hosts Mark van Atten and Leon Horsten. I'm getting exceedingly little input from the boys, maybe 5% of the paper, so I feel it would be okay to use it as my own. The pay-off in the paper is based on Nick Herbert's little essay "Quantum Tantra." I do need to see if Nick minds if I use his idea. Conceivably I could collaborate with him, but that would be more work I think.

(Thesis) Upon introspection we feel there is a residue that isn't captured by any scientific system; we feel ourselves to be quite unlike machines. This is the sense of having a soul.

(Antithesis) But (a) the work in AI and simulations, (b) genetic algorithms, and (c) simple considerations of degrees of computability seem to indicate that any clearly described human behavior can be emulated by a machine. Where is, then, the missing soul?

(Synthesis) The "soul" can be given a scientific meaning as one's immediate perception of one's uncollapsed wave function, particularly as it is entangled with the uncollapsed universal wave function of the cosmos.

(Two possible conclusions) Either (a) machines, qua physical objects, have uncollapsed wave functions as well, so they too have the same kind of "soul" that we have or (b) there is something so far unique about how we manage to couple our soul experiences with our logical reasoning. Re (b) it would be worth looking into quantum computation as currently formulated.

In other words it's time to play some catch-up ball re. quantum computation. Well, right now I have tons of time, so why not read up on Quantum Mechanics again. I found my nice old intro by Daniel T. Gillespie *A Quantum Mechanics Primer*. I bought it at Rutgers and read it in Geneseo. On the back pages were written a couple of drafts of 1976 poems, from Geneseo days, "Drunken Hearted Man," and "She Got a Phonograph." I also bought a new book on *Quantum Computation and Quantum Information*.

I finished reading the *Primer* on the beach at 14th Ave. in Santa Cruz yesterday. It was so lovely, there, lying beneath the tall Monterey pines, with the long luscious crunches of the clean green waves running a hundred meters parallel to the beach.

More and more I'm fascinated by the idea that my state of mind is exactly like a quantum state, a wave function *me* that is a complex-valued function over an n-dimensional space with n quite a large number describing my degrees of freedom (the log of my number of mental states?) If A and B are observables, this means they are questions you might ask me. If A has answers a1 and a2 and B has answers b1 b2 b3, then there are corresponding eigenstates Ai, Bj. When I am in eigenstate A1 I for sure believe the answer is a1. But normally I'm, like, 0.7exp(iu1) A1 + 0.7exp(iu2) A2. I'll go into this at greater length outside the journal notes, it's too mathy and too preliminary for this venue.

### *June 11, 2003. Discover Memoir Notes Once Again.*

I remembered that at Esalen I flashed on this a book idea: *Memoirs of a Crazy Mathematician*. And then I once again stumbled onto the Memoir Notes document I'd made in April and forgotten once before. I only found it because while spell-checking I noticed I'd made a dictionary file called "memoir.dic" and I'm all, huh? My mind is like a sieve, maybe the sooner I do memoirs the better. I'm gonna integrate the Memoirs notes into this document.

*The Answers: Memoirs of a Crazy Mathematician*. That's very commercial. And everyone but me thinks I'm "crazy" anyway. So why not cry over the insult — all the way to the bank. But no way do I in fact want to call myself crazy. Just for a few bucks? No.

Yeah, man, I owe the world a memoir. Seems like every week I see another memoir by a 25-year-old in the N. Y. Times Book sections. Saw another in City Lights yesterday, something about a kid who met Feynman. You pick that up right away. The layperson is more comfortable with a memoir.

Eventually I'd like to do a novel about a crazy mathematician as well. His "craziness" could involve some of the ideas in *The Answers*. But switching to this project wouldn't solve my current problem of how to write a commercial non-fiction book summarizing, in particular, what I learned in Silicon Valley over the last 17 years. I would prefer to first do the non-fiction book and then eventually the novel.

***June 14, 2003. Thinking About Computers.***

I walked around North Beach the other day, making notes. And keep folding old proposals into what I have here. I got it down to two main lines: either a Big Book Of Computers or Memoirs.

And now today, I decided to set aside the Memoirs notion for now and go ahead and try and finally put a stake in the Tome's heart, this project that's haunted me for going on fifteen years. Just do it.

Why didn't I do this before? Well, whenever I'd show it to my science-fiction agent, she'd be kind of negative about it. And, frankly, when I've discussed it once or twice with editors they haven't been that interested either. Particularly when I mention writing about chaos and fractals, for instance, they feel like it's been done. But, lord, how many books are there about Relativity? These things are here to stay.

Still can't fix on a title. My persistent illusion that the right title will snap it all into focus as a fresh new book.

***June 20, 2003. Idea for Story/Novel.***

Qubits. Netflix downloads a superdense coded info pattern via an entangled qubit and a guy gets the wrong movie, it's the true story of his life.

Air protozoa.

Have UFOs shaped like the medusae, siphonophores, etc. in Haeckel's amazing TK. I picked up a copy at the Monterey Aquarium last week; Sylvia and I were at her "Young Rhetoricians" conference there for four days. Haeckel's last work was on crystals, I have to find out more about that, the note in the book says it had to do with "artificial life" !?!

***June 26, 2003. Computers and Reality. Brockman. No Equations.***

Ever since North Beach, like two weeks, I've been working on my book proposal. I wrote the big science-book agent John Brockman about it the other day:

"I'm putting together a proposal for what I hope will be a big non-fiction book with working title COMPUTERS AND REALITY. My present idea is to cover all the good old (chips, fractals, chaos, AI, computability) and new (Wolfram's work, the web, quantum computation, videogames) computer things in a non-technical way with a focus on how these ideas can help the average person live a richer, more enlightened life. Working with computers has changed the way I see the world in some interesting ways — and I want to share that. I'll tell you more about it later on."

He answered, "COMPUTERS AND REALITY sounds good — and it's right up my alley."

How refreshing to get an answer like that!

I was looking at this book by David Deutsch called *The Fabric of Reality* (which kind of validated the use of the R word for me), and I noticed that his book, which I think sold pretty well, doesn't have any displayed equations or computer code at all. I wonder if I could write my whole book without equations, code, or even subscripts. If something requires technical symbolism, then just don't put that in. This would mean, however, that I couldn't explain, for instance, how Rule 30 works, or show my nice clean CA wave equation Wave(C) = NeighborhoodAverage - OldC. But maybe I could live without them. (If not, I could banish the equations to footnotes — but then average people would know they were there and still be scared. Maybe put them on a website? But why go to all that work for something that isn't even part of the book? But the geeks could find it then. Could have a website with downloadable programs, Java applets, and the hard-core symbolic stuff in PDF.) But I'm getting off the point, which is that it would make the book more commercial to have no equations. I think it might be doable.

I'm on an airplane now with Sylvia, flying to Boston to attend Wolfram's first NKS (*A New Kind of Science*) conference. And then on to Greg, Karen Johnson, Maine, DiFilippo, Block Island, and Readercon.

***June 27, 2003. At the NKS conference.***

Brian Silverman says call it *Computation and Reality*, not *Computers and Reality*, but right now I'm thinking, no, it's the latter book I want to write. I do want to include computers as media machines (games, the web) and not just talk about nature as being made of computations. He doesn't like my book plan, I (perhaps incorrectly) imagine he's a little miffed that I'm not writing the book with him as we'd vaguely discussed.

I have to go present at a panel on NKS in Higher Education now. Looking back at the Leuven syllabus, I'm thinking I should maybe use those chapter titles after all. Each followed by a colon and what's *actually* in the chapter, though.

***July 2, 2003. Wolfram's Input.***

I saw Wolfram this week at the NKS 2003 conference. He took me aside and urged me to write my next book all about NKS. And I'm tempted. His NKS sold, he says, 300,000 copies. And people might want a shorter simpler version. I can visualize a sleek, slender volume called, say, *On Computation*.

But to just do that would be subjugating myself. I want to express *my* ideas about computation, dammit. There could possibly me a middle way, of sorts, that is, to bring NKS repeatedly into the book, relating it to all my chosen topics.

Nobody seems to like my present *Computers and Reality* title all that much. Maybe I should go with *The Lifebox*. That's "my" new idea, that old lifebox thing. So it could be the payoff summary of what the book's about. Or *The Lifebox and the Quantum Mind*, which would be more inclusive. Maybe I should drop some topics and focus on those two. In particular could I drop VR and computer games?

I'm planning such a radical revision of the notes that I think I'll save off my current outline as "Computers and Reality Notes, July 2, 2003.doc".

### *July 4, 2003. Lifebox, Quantum Mind, NKS Triad.*

Thesis: the lifebox.

Antithesis: the quantum mind.

Synthesis: NKS-style automata, which possibly are quantum computers.

Maybe just call it *The Lifebox*.

I see the mind's churning as being like the eddies in a von Karman vortex street bouncing off each other. I love Wolfram's notion of coming up with a higher-order automaton procedure involving the eddies. In this way we are free to ignore sensitive dependence, that is, we can ignore (a) the thermal bath randomization and decoherence, and (b) we don't have to pay so much attention to the excavation of digits.

### *July 14, 2003. Popular Interest in AI.*

Yesterday I was on a panel about "The Singularity" at the science-fiction Readercon 15 in Burlington, Mass.

I was surprised, again, how really naive most people are about AI. Always you hear the same arguments why a machine can't be like a person. That a machine "doesn't care" if it wins (but it does "care" if we give it a utility function), that a machine "doesn't make mistakes" (not only can random errors easily be produced but, in use, machines tend to accumulate dead storage areas that make them run less well), etc.

As always, the faces light up when I talk about the merged higher consciousness that we know, from the inside, that we have.

My fellow SF writers seemed wholly unable or unwilling to discuss the Singularity argument's three moves as formulated by me: (a) strong AI will occur, that is, machines equivalent to humans in mental power will be evolved, (b) once we have strong AI, we can easily get superhuman intelligence by running the machines faster and giving them more memory, and (c) each generation of superhuman machines can design a still smarter next generation, setting off a cascade of more and more powerful artificial minds.

Of these three steps, only (b) is unexceptionable.

(a) may in fact never come true; strong AI may forever remain a will 'o' the wisp. At this point really the only strong general purpose method for AI we have is neural nets. Training a given net isn't terribly slow, but trying to evolve towards the correct net architecture is an exponential search problem.

(c) is by no means a given. For if we look at the first step of the cascade, in which we humans design machines able to work as well as (let alone better than) ourselves, we find that this step is an exponential search problem. So why should it necessarily be the case that machine generation n can very easily design generation n+1?

### *July 23, 2003. Brockman Enters the Fray.*

So I'm signed on with John Brockman; he only wants to represent this one book, doesn't want to touch an SF novel, we have a single-book agency agreement. Though he thinks he'd be interested in a non-SF novel. I'll probably go back to my usual agent after this one.

I certainly do want to write SF novels again. Well, it's always just one book at a time, anyway, isn't it.

He doesn't like the July 17, 2003, version of the proposal, but thinks I can get there. Says it needs to be — *unpacked*.

He didn't like *Computation and Reality* as a book title, said the publishers think computers are dead, what with the dot com bubble burst. Said he's just failed to sell a book of essays (edited by him) called something like *A New Humanism*, about the theme that everything is about computers these days.

Most of his book title examples had colons in them, like *Einstein's Space and Van Gogh's Sky: The Blahblah of Whatever*. I'd like to get away without a colon.

Didn't like the chapter titles, "Our Electronic Servants," "The Web of Knowledge," or "Consciousness." Used the phrase "shopworn."

He said, get this, "You're in California — score some dope and go crazy with it. Take the readers on a trip. A ride." "I can't get dope," I said. "I don't mean it literally," he backpedals. "Make the book fun, exciting, take people to a new place. Make it trippy."

He didn't like the Preface draft getting into Reality and :the world, society, and the mind. Too familiar-sounding.

So I rewrote it and changed the title to *The Lifebox, the Seashell, and the Soul*.

This is all very welcome. Instead of, "Rudy, I don't think people are buying this kind of book anymore. Why don't you do a book like that book about longitude?"

### *July 31, 2003. Brockman Pushing Me*

So Brockman liked the second version of the proposal, said I'd convinced him, wanted just some little tweaks, said we were almost done.

Said he doesn't want me to mention my SF career very much, in fact every time I talk to him, he dumps on SF, which is of course tedious for me, having to be polite in the face of that kind of snobbism. "Marvin Minsky writes SF," he tells me, "but we don't talk about that to the nonfiction editors." Rrright.

The assistant Fed-Exes me three samples of successful proposals, which are sort of helpful, and sort of not. More and more I begin to feel that I have nothing new to say. I skim them quick, then send a tougher, weirder Version 4 proposal, something closer to what I really want to do, punker and using more made-up SF words, more in your face. Maybe that's good, maybe the more they needle me the better the proposal gets, the closer it gets to being the real me. Or maybe the proposal is now just fully beyond the pale.

At moments I think maybe I really don't have anything commercially interesting to say. I'd hoped he'd call back today, but now it's too late, and I'm going camping with Jon Pearce tomorrow, so I won't hear anything till Monday. So all weekend I can imagine the situation is hopeless. Why do I get into these things?

### *August 4, 2003. Openers.*

Still no word on proposal Version 4. My mind turns to plans for Version 5.

How about break each chapter into (a) self-aggrandizing name-dropping reminiscence, (b) quick factual survey, (c) how this affects your mind and your life in your

American home, (d) a minifiction.

I have begun thinking about some bragging memoir-style openers to start chapters or sections with. Here's a few candidates, classed by the chapter they might appear in.

1. "I recently finished a novel about the life of the Old Master Peter Bruegel. In some ways, writing the book was like spending two years in the Sixteenth Century. One crashingly obvious fact that I came to know right down in my bones is this: *human nature doesn't change*."

2. "As a young man, I spent several afternoons discussing philosophy with Kurt Gödel..." Re languages, "As Bill Gosper used to say to me..."

3. "When I first met Stephen Wolfram..."

4. "When billionaire John Walker carried his Mandelbrot set machine into my house I was impressed..."

5. "I'll never forget the afternoon that I went to Tim Leary's house in Hollywood, took apart his computer, and installed my CAM-6 board so that he could look at cellular automata."

6. "When I was channeling Jack Kerouac in Tucson at the Genomics conference..."

7. "My old writer friend William Gibson has an amazing notion: 'the universe is created without regard to human notions of "outcomes," that events are patterned in their own way, and fear and expectations are just an aspect of seeing it through human eyes.'"

8. "Ted Nelson's Xanadu friend with the zit on his nose..."

9. "Nick Herbert is ..."

### *August 6, 2003. Brockman on Version 4.*

Today I got some delayed email.

"Getting better, I remain concerned about the page and a half proposal ("summary"). There's not enough going on to inspire publishers to pay the kind of money we both are looking for. And I worry that too much is tired and superficial (i.e. discussions of computer as word processor) and not going anywhere. How about slowing the process down, taking a week and going deeper and longer.

"My comment re: physics, which Russell passed on to you was derived from the fact is that all the key players (Lloyd, Wolfram, Deutsch, etc.) are all physicists and not computer scientists. Computation is physics. In that regard, the focus on "computers" come off as lightweight, but when you focus on "computation" you hit your stride. Also the above mentioned are a cutting edge of sorts and publishers are interested in "new" and "next". To the extent that you position this as a next step following Wolfram's book, you have a winner. But the discussions about freeing oneself from the computer are not going to fly in the marketplace."

And then I phoned Brockman, and he said about the same stuff. Also that I shouldn't be turning around my new proposals so often, I should think more, take my time, don't be lazy, don't rush it.

It's really an issue of fear. I'm afraid I have nothing to say, or that I can't cast my ideas in a commercial form. I'm afraid to work hard on the proposal, afraid to change, to rethink my ideas, afraid that the harder I look into myself the less I'll find there.

Well, I'll spend a few days on Version 5.

***August 11, 2003. Version 6 is a Go!***

So I made a nice version 5, and today Brockman's assistant Russell Weinberger called and said they liked it fine, modulo two changes, drop the too-slobbering second paragraph from the start of the proposal, and go back to an earlier title.

I had made the title *Seashell Soul: Enjoying the Computational Worldview*, and they had me roll it back to *The Lifebox, the Seashell and the Soul* with no subtitle. They thought *Enjoying the Computational Worldview* sounded "too geeky."

So I made those two small changes to have a Version 6, and now we'll see what happens next. I wonder if they can really get me a large advance. If so, I think I'd be pretty stoked to write the book. I guess I could start on it pretty soon. Maybe I'll begin by thinking about the history of computation once again, but this time without focusing on machines, allow things like canals to be computations as well.

***August 19, 2003. Meeting Brockman in NYC***

I met John Brockman, his son Max, and Russell Weinberger for lunch in NYC on August 19. Brockman turns out to be older than I'd expected. But, duh, he's my age. A lively character, certainly, and I found it easy to talk with him. And he seemed to enjoy me.

He sent the proposal to about ten good houses, all at once, and seemed fairly confident that one of them would give us a good advance. "I wouldn't be having lunch with you if I didn't think that," he said. "Max is expensive." His office was impressive, a penthouse in a building on Fifth Avenue across from the Plaza Hotel.

He did send it to Four Walls Eight Windows, kind of an ace in the hole.

Simon and Schuster has already turned my proposal down. John didn't think we'd get any action till September, when everyone's back from vacation.

***September 4, 2003. History of Technology as History of Computation***

Now I'm back to teaching at SJSU.

While I'm waiting to hear what the publishers say, I started working on the nine between-chapters short-short stories I wanted to put in the book. I finished the first one, "The Kind Rain," and am starting on the second. The first was a thousand words, which seems just about perfect. Even if the book doesn't fly, I can use the stories somewhere.

I need something to write in any case, with the *Frek* revisions all done. I feel like I'm wasting my life if I'm just teaching and not writing. I always find it hard to start a fresh story, much harder than having a novel to keep working on, so in a way it's good to have the discipline of trying to write nine one-week stories in a row (optimistically assuming I can do one a week). Maybe to finally get the knack of the short-short story form. I always used to love Frederick Brown's short-short SF stories.

I've been reading a lot, too, boning up on computation stuff. I printed out three or four papers by David Deutsch and some other quantum computation characters. He seems less and less sanguine about large quantum computers being physically feasible, let alone human brains actually being quantum computers. I'm thinking I might after all do a limited modified stations-of-the-cross thing [cf. the Nixon White House Haldeman-Erlichmann conversation about considering a "limited modified hang out" on Watergate rather than a full "let it all hang out" (glorious Seventies expression, that)], like talk to Bennett, Toffoli, Margolus, all of whom I know from the old CA days, and maybe even hook up with Deutsch in England. Could of course talk to Wolfram, too.

I started reading *The Bit and the Pendulum*, a fairly low-level book by a science journalist. He starts right in with an idea I'd been mulling over, that the computer is a popular machine whose presence has changed the mental paradigms we use. He mentions the clock and the steam engine as two earlier mind-changing machines. But I can't go on reading this book, the man writes down too much, he over-simplifies.

I'd like to do a bit more with the history of technology. The obvious move would be to say that each technology had it's own particular concomitant world view. But I think it would be more fun to start out from a hard-core automatist stance, and say that each new technology was fundamentally an additional form of "computation" in the broadest sense of the word. I love this kind of McLuhanesque play.

Last night I was thinking about it all night in my dreams.

| **Technology** | **Computation** |
|---|---|
| agriculture | Seed produces a plant. Animals make animals. |
| spinning and weaving | Fibers make yarn, yarn makes fabric. |
| smelting | Heat turns rocks into slag and metal. |
| clocks, locks, etc | System of gears does the same thing over and over. |
| steam engine | A "living" or autonomous machine that eats coal. |
| the dynamo | Subtle invisible energy through wires. |
| radio and television | Subtle invisible waves through air. |
| atomic power | Treating the atom like a machine. |
| computers | Complicated repeatable patterns made of electricity. |
| biotechnology | Tinkering with the genome. |

***September 6, 2003. Selling the Book***

Talked to Russell Weinberger. John Oakes of Four Walls Eight Windows has made an offer for Four Walls Eight Windows, but many have turned it down. Russell sent me some of the rejection letters, which were dishearteningly blind to what I'm planning to do.

I thought of the phrase, "I went to the demonstration, to get my fair share of abuse," as Mick Jagger puts it in "You Can't Always Get What You Want."

***

Yesterday afternoon, about two hours after thinking of Mick's phrase "I went down to the demonstration to get my fair share of abuse," I turned on the car radio and heard, yes, *You Can't Always Get What You Want* right from the start. Synchronicity! Hearing it I realized, indeed, I can't always get what I want, but if try sometimes, I just might find, I get what I need. *How true*.

Also I thought about why "get my fair share of abuse" rings so true for me. My projects have always been "greeted with howls of execration" (quote from last page of Camus, *The Stranger*). If people didn't question and disagree with what I want to write, then I wouldn't be the same old Radical Ru. I'd be a sell-out. Even though I'm always imagining I'm going to sell out, I never seem to be able to figure out how.

Actually, it'll be good to work with Oakes again. He'll be likely to give me carte blanche on the writing, which is always nice.

What is a little deflating is to not score some huge advance. But I really do want to write this goddamn tome, I think. My (latest) White Whale.

Certainly there would be some of my loyal readers, now and in the future, who would be wistful if my fabled computer tome never got written. Including me, I guess, me as reader of my own work.

If I don't do this, then what?

### *September 15, Email to Greg (on Gratitude)*

[Greg Gibson is a writer friend who tends to spend years crafting his writing proposals so as to garner a big advance.]

Well, I'm gonna just write this book for a modest amount that seems to be as good as the proposal will bring in. Having some offers is better than none, and I've seen quite a few of my writer friends in the "no offer" bin these days. I'm working on generating the emotion of gratitude, always a good one vis-a-vis life. The pub is Four Walls Eight Windows (4W8W), who published my collections *Seek*! and *Gnarl*!, also reprinted *Hacker and the Ants* and *White Light*. I see their books in all kinds of stores, they're a notch above small press, I'd say. Owned by the one guy, John Oakes, which makes them more spontaneous in their decision-making.

A difference between our situations is that I have this one specific book that I want to write, and these days you're more after the eidolon of the book the publishers want. I currently have a kind of odd working title, *The Lifebox, The Seashell and the Soul*. The feeling being that publishers these times don't want to hear about computers. Of course they figured out my book *is* about computers, and that it's wide-ranging, and that I'm a weird old man, that I don't see rosy futures or (flipside of banality) view with alarm, that I'm left coast, the usual problems. Now that they've seen the proposal, it's not like I can change the slant and send it around again, not as long as it's still fundamentally the same book.

It'll be about what I've learned and speculated about computers in the last twenty years, with the anticipation that, in the process, I'll reach some new insights and higher integrations. Like any other adventure, I don't really know in detail what I'll see before I set out.

Writing this letter to sell myself on the project as much as you...

I'm ready. Rearranged my bookcase, moving the new ref books into place. Life's empty when I'm not carrying a foetus in my brain/womb. No little kicks to the medulla. I can't go without my endorphinic fix no longer.

Meanwhile: What *is* a computation? (Section 1.1).

Sign me, Grateful Writer

### *September 16, 2003. 4W8W Again.*

So now Oakes made a final offer, better than I got for my last novel at least, and it's as high as we'll get. Russell is letting him keep English rights.

### *September 21, 2003. Anxious.*

I feel anxious about having enough stuff for the book, or if it's worth writing. If none of all those publishers wanted it, how good is it? Who cares about the computational worldview?

On the other hand, I don't really have anything more pressing to do. And I've always wanted to try this. So soldier on, Ru, soldier on. Make it your book, do it your way, tell a story that interests you, and you'll have something worth doing.

***October 4, 2003. Getting going.***

So now I'm working slowly on Chapter One. Lots of distractions with my four-day-a-week teaching schedule and the need for entertainment on the weekends.

But last night I recalled how ahead-of-the-pack I felt at the NKS 2003 conference. I think nobody is better equipped than me to write this book. I'm at the edge, which is why it's hard.

Working on Chapter 1, the lead section, "What is a Computation?", and trying to get up the chutzpah to tackle the messy problems of analog computation.

***November 1, 2003. Back Into It.***

I've been away from the book for the last few weeks. First I had midterms and a lot of homework to grade, and then I got the copy-edited manuscript for *Frek and the Elixir*, and worked on that for a week.

I had seriously decided to change the title to *Geek Philosophy*, partly at my friend Marc Laidlaw's urging, but John Oakes begged me to stick with *The Lifebox, the Seashell and the Soul*. He says it doesn't matter that it's complicated to explain the title, what counts is that it's intriguing. And, hey, he's the guy giving me the money, and Laidlaw hasn't sold a book in a number of years, so who are you gonna listen to?

***November 18, 2003. Moving Right Along.***

So now I've got 20,000 words done, whoah, and I'm only 2/3 through the first chapter, so it'll be like 30,000 words in Chapter One. I beat the table of contents down to five chapters, and I hope some of the later chapters will be shorter. I'd prefer not to go over 100,000 words. Maybe I better get it down to four chapters. So it might be 120,000 words. Okay, I just revised my working outline down to four chapters. That means, wow, I'm already a fourth done!

This mamma is writing itself. Just like *Spaceland*. It's like all I have to do is type it down while the voices in my head tell it to me. They don't tell me a whole lot at once, just a little every day, but it's accumulating as fast as snow on the ground in a blizzard. I have so much to say about all this stuff. The nice thing is that with so much material compared to the space I have to fill, I don't have to write about stuff that I don't really care about. Toss the history of programming languages overboard! Good-bye to the worries about biotechnology! So long, Moore's Law! Sob! Well, maybe I can still talk about those things.

I think, though I should go ahead and try and finish topics off as they come up, and not be mentally planning to really do it right later on. I might not get back to some of the topics at all.

***December 8, 2003. Done with section 1.3, The Physical World.***

I'm finally done writing the little survey of physics. I'd been worried about the quantum mechanics section, but I think it came out alright. Hopelessly dated? I didn't mention irreversibility. Oh well. I've always hated irreversibility. Just write it and mail it in, Ru. Get it done.

I need to get back to the main argument of the book. In getting so deep into the physics, I kind of lost the thread. Better reread the eighty (!) pages I've written so far and remember where I wanted Chapter One to go. Next up is supposed to be a section on Life, with subsections on Nature, the Mind, and Society.

Yesterday I was walking in the woods. Start there?

### *December 11, 2003. Contracts!*

I finally got the 4W8W contracts from Brockman. He's really been working it over, says Oakes, "changing the most unexpected things." But they're done and they look good. The money's not so bad. I taught my last class yesterday, so hope to make some progress in getting Chapter One done. Well, finish it by end of January, anyway.

Worrying about finding more and fresher computer examples for my section on Life. DNA ~ fractal parameters. Reaction-Diffusion CA ~ Development. Feedback system like flocking boids or balancing unicycle or getting the parameters of a computer game to work ~ metabolism. Wator ~ ecology. Blind watchmaker CA program ~ evolution.

In a way there really aren't so many computer examples. But I feel like there are more than I can put my finger on.

### *January 13 - 14, 2004. What To Say About The Mind?*

My mind is a stream with ripples I call thoughts (or maybe it's something else) and (if it's like a stream) all I can do is throw pebbles into the water.

I've been pushing for this triad notion that the lifebox and the soul are bridged having the lifebox run an activator-inhibitor rule (like the ones that generates the cone shell patterns or the leopard spots or the bones of your hand).

Today I thought some about how to organize this, and in the process I'm having some doubts about the particular applicability of an activator-inhibitor rule (I'll call it a RD rule for reaction diffusion, I can't use AI as this would sound like artificial intelligence.)

I see a series of levels.

**Mind level 0: Quantum mind merged with cosmos**

Merged with the One. The Herbertian quantum mind. The sense of the World. Quantum mind is really only a metaphor, in connection with the brain, but we'll see that it has a certain validity in the esoteric field of quantum computation.

**Mind level 1: Neural net reflexes.**

Neural nets to control reflexes. Like two photosensitive spots connected to a two neurons, each connected to a bending muscle which is also connected to skin muscle. Some simple weights on the sensor inputs. An ability for the neurons to be refractory so they don't fire all the time. Well understood standard issue neural net type stuff. Think of Genghis the Ant. Note that neural nets aren't dynamic, not like thoughts.

**Mind level 2: Ideas as patterns of neural activation.**

Now comes a somewhat mysterious level at which we form basic ideas. I'm talking about simple facts like “this ball is red,” or “ head is hairy.”

I see ideas as being patterns of neuron firings, like static loops, at least in the short-term memory. In the long-term memory we probably have something less volatile, like stable geometry of axon dendrites or at least maybe memory biochemicals. But the short term idea is probably electrical.

(2a) *Patterns of activation and inhibition.* To keep something in mind, you keep circling around a little loop or knot or subnetwork, activating perhaps a series of sensory memories blue-hot-hungry-sweet-wet. Maybe inhibition plays a role, in terms of not letting the thought energy branch off into other connections. Think of sand moats with water sloshing around in a circle, the sand is inhibition, the water is activation.

This can arise a limit cycle of a process.

What controls the motions of activation and inhibition? Could it be an BZ-style rule? One difficulty here is that the activator-inhibitor or reaction-diffusion rules are formulated in terms of CA spaces, and the space of brain neurons doesn’t have that type of structure. It’s a network.

But perhaps we could run such a rule over a network. Visualize a 3D CA running a BZ rule and have the cells be nodes connected by strings to their neighbors and stretch a bunch of the strings and scramble things around, maybe drop the mess onto a tabletop and the BZ is still happening, but on casual observation you might not be able to *see* the scrolls. In the same way, the network of my brain cells can be running an activator inhibitor rule even though no type of brain scan would reveal scrolls. The spatial arrangement of the neurons doesn’t match their connectivity. Actually there probably is *some* match, so one might expect to see traces of scrolls in the brain activation patterns. [Technical problem: a network doesn’t have the neighborhood structure of a CA, like if A connects to B and A connects to C, that doesn’t imply B is “near” C on a network, thought it does in space. Maybe I could set up an N-d space of some kind to have linked things close? The i-th axis being (1-strength of connection to node i). This is too hard to think about.]

Now what is the stuff that spreads the activation and inhibition? Rather than thinking of it as concentrations of chemicals, I could, rather think of it as excitory and inhibitory electrical signals. Suppose each neuron is in a state between -1 and +1 (inhibited vs. excited) and suppose that each has axons leading to other neurons, and that these axons transmit a -1 to +1 signal matching the current activation level of the cell.

One thing that makes me pause here is that in Turing stripes and BZ rules, we have the activation and inhibition spreading at different rates. But it seems like an electrical signal be transmitted at the same rate whatever its value. But a neuron could respond at different speeds to different kinds of inputs, might move more slowly into activation and be able to drop abruptly into inhibition, thus effectively giving different rates. In this connection think about things like the Hodge rule or even Brian’s Brain which mimic excitation and inhibition without even having two substances, they get by with simply a range of states.

If the process is something like a CA running on a network of neurons, we can ask what is the rule in each neuron. I’ll suggest a complicated and a simple idea. The complicated idea is to have different rules at each neuron. The simple idea is to have the same rule at each neuron.

(2b) *Complicated neuron rules.* I think here of a Stuart Kaufman network. Imagine having N cells, each with k links to randomly chosen other cells, and give each cell a random Boolean function F used as NewVal = F(k inputs), and just put a 0 or 1 in each cell, and update everyone in parallel a few times, and I think in the limit Kaufman commonly found limit cycles of a size maybe square root of N. [Can one make Kaufmann network visually interesting?]

(2c) *Simple neuron rules*. This would be in the Wolfram style of making rules as simple as possible and finding the behavior you want anyway, without having to put in a lot of randomness into the system to start with. Make a network of N neurons each linked to k others, but use the same Boolean function at each node. Wolfram would do a search over all the possible Boolean functions and no doubt he'd find the four canonical computation behaviors: dying out, periodicity, seething, and complexity — the behavior depending on which rule you picked.

Could we simplify further? How important is it to have random connectivity? What's wrong with simply using k-neighbor CAs? And then we really *are* just talking about CAs, as having the same Boolean update method for each node means stating a CA rule. [To go more towards network, how about giving each cell A its nearest neighbors NA plus one remote neighbor rA. And say we pick the remote neighbors so they aren't near each other. That is, if A and B are close then rA and rB aren't close. Maybe there could be a nice canonical way to do this. If I wanted to program this, I could have a CA with an extra real number field dist, and it would take the usual nearest neighbor inputs plus an input from a cell whose index is dist*N where N is the size of the CA array. I could fill the dist fields at startup with a randomizer.]

**Mind level 3: Patterns of facts.**

Key mental notions that we form are the following. Each needs its own kind of explanation.

- Notion of objects and actions, nouns and verbs. This could evolve.
- Notion of self. The videogame notion explains this pretty well.

**Mind level 4: Lifebox map of what I know.**

I can statically map all the facts I know. Everything I know can be made into a web page, like a blog, an info dump.

As well as the anecdotes and memoirs, I (or a program) can provide a rich set of links connecting the anecdotes. Lifebox is like fossils of a life form. Traces. What makes the traces?

To make this a full-fledged lifebox, we supply a *browser* which fakes understanding natural language well enough to return reasonable links to database elements in response to spoken or written questions. We also incorporate an "I'm feeling lucky" feature to have the lifebox pseudorandomly volunteer information. The lifebox can't pass a true Turing test, as it won't understand or remember things about the interlocutor, but in a weak sense it will pass it, the weak sense being that you simply ask it questions about itself. Like having a conversation with William Gibson, who's been interviewed so much that, no matter what I say to him, he responds with an interview question answer, rarely bearing any relationship to anything about my own life which I may have been trying to interject. More precisely, whatever I say is combed for trigger words to access Wm. Gibson lifebox info. And of course I do the same to others, when I can get away with it.

**Mind level 5: Artificial intelligence.**

Now we want to add something to make the lifebox have ideas.

- 3D space to fit objects into.
- Time with before and after.
- Grammar

(5a) A neural net can memorize a lifebox, but it evolves simply towards efficiency, doesn't really figure anything out. Like a minimal code, really. It can extrapolate a little, that's why training handwriting-reading neural nets works, e.g. But you don't expect a neural net to come up with new thoughts..

(5b) Logic as a tool for reaching new facts that you can't see, or that haven't happened yet.

(5c) Grammar as logic-like tool for producing coherent new utterances.

**Mind level 6: Creative thought.**

Here again I think we must turn to complex computational processes, focusing on intrinsic randomness, like in (2). The phrase "train of thought," is suggestive. In this zone we must be outside rules, I think.

A highlighted set of ideas to indicate the lambent spotlight of consciousness.

An abstract thought a hyperlink connection of more than three elements. Any two memories can always be connected without much effort, to bring in a third makes it a thought.

***

Activator Inhibitor Rule or Some Other Class 4 Computation

Browser

Links

Blog

Grammar(Many, Objects), Logic(Spacetime, prognostication), 3D Perception (One, World).

***

And none of this even touches Nick Herbert's mind as quantum system line of thought. Is the collapse like a BZ scroll being squeezed into a tubule? The empty mind, merged, distinctionless, is this like the mind of a dog? Pure attention, pure self, pure now.

The quantum mind is a state of generalized activation and no specific bright spots. Zen mind. Empty mind.

***

And does twinking fit in?

***

Tired in bed, the thoughts aren't so fun, they are in dull loops. Class 2. Classic example, remembering a fight: I said, she said, I said, she said, playing it over and over.

Class 1 would be even worse? Monomania. Though looked at in another way, yogins admire one-pointedness of mind.

Thinking of the mind the next morning, walking around. The miracle of all this being done by a carpet of cells that have grown themselves into a mat. How can it be? I see the footbridge, I have a model of it, walk onto it, look down at the ripples in the stream, the standing waves of foam, thinking these are like the mind, a woman walks past, her face twitches anxiously to see a silver-haired man in black jacket and sweatpants on the bridge, I think of Joseph Campbell, of myth, the troll under the bridge. All this coming out of the meat weave in my head. The associations somehow kept alive in the background, ready to pulse at my call, and always the new associations. Class 4.

I think it's very rare we have Class 3 thought patterns. A chaotic storm. Like maybe when I took acid it was like that? Not even. That was more like Class 4 very dense and fast. Though maybe the White Light, or any milder type rush is class 3, with all the neurons firing there's no real content, simply a sense of stimulation. Any connection here to why stupid people (athletes, businessmen) like coke?

### *February 8, 2004. Iced In.*

The start of the school semester always reminds me of the analogy I thought of when I was working on *The Hollow Earth*: A ship expedition headed for the Antarctic getting frozen into the ice, has to wait a season till it can move forward. Well hopefully it won't be a whole season, but I haven't made much progress on the book this month. I did reorganize it; Chapter One was getting so long that I broke it into five (!) chapters. I'd thought initially of going over everything fast in Chapter One as warm-up, then doing the more technical Wolfram stuff, then going over some examples in detail. But I'm hitting the Wolfie stuff stronger earlier on and doing the example stuff right away, and after that I'll do the more technical stuff. Idea being that it's never a good idea to hold back from the reader, go ahead and show your best stuff as soon as you can.

I've been reading some interesting books. Rodney Brooks, *Flesh and Machines* is terrific; it has so much good stuff on the Mind to use that I want to go over it and type notes into this "Notes" file. Stephen Johnson, *Emergence*, has some usable ideas as well, but the guy is really an English-major type and doesn't actually get a lot of what he's talking about. I'll make notes on this one as well, I think. He has useful stuff on Society in particular.

Before tackling the somewhat daunting section (now become a chapter) on the Mind, I wanted to go over all the earlier stuff and smooth it out, and I'm still in the throes of that.

### *February 25, 2004. Zhabotinsky Scrolls.*

I'm still pretty much iced-in, haven't really done jack on the actual writing of yon tome (i.e. *The Lifebox, the Seashell, and the Soul*.) I've been in mega-hacker mode, revising my Pop computer game framework for CS 240: Graduate Software Project and writing some demos of lighting and textures for my CS 116B: Computer Graphics II. Papers to grade as well, but mainly its the hacking that sucks up all my time. Once my colleague Michael Beeson said, "Hacking is like drug-addiction. It uses up so much of your time that you never get around to doing the normal maintenance things like eating, washing, answering your mail." The other day Sylvia came home at 6 PM and I was still in my PJs, hadn't eaten all day. Debugging the Pop framework. Told my students about it, and they seemed to appreciate it. The kind of thing they'd do too.

I wonder if I can talk about what hacking is like in my book. And I'd like to get into how teaching OpenGL lighting has enhanced my appreciation of the specular highlights on things (staring and staring at Sylvia's dully opalescent toroidal silver brooch in church on Sunday), the diffuse and ambient lighting, the emissivity, the radiosity. Not sure if I can work any of this into the book. Actually today, I really was kind of having fun with the hacking, though at some point there's always a bug that takes way too long to fix, and your butt begins to get numb on the chair. Nobody can ever really know how much fun I do have. When I'm just doing stuff and not writing in my journal or anything, my life has this pure, unobserved quality. Hacking and light.

Maybe both could fit in *Chapter Seven: Waking Up*. An example of not being awake could be the blood-lust hacking frenzy, an example of beginning to wake up is noticing lights in a new way, and the best is to go outside on the rain-washed hills like I did today. There's one spot on the path near the former nunnery at St. Joseph's hill, it's so bosky and closed in, the oaks so tumble-down and gnarly, a real little Eden, I'm always so thrilled to be there, and at the same time filled with a kind of anguish that I'm not there all the time.

The last few days I've been cranking to get good Zhabotinsky scroll images. Partly for the book, particularly the sections on morphogenesis and ecology, but also because Wolfram is planning to phone me on Tuesday and try and do some experiments together over the phone. By way of getting me ready, he's mailing me a new copy of Mathematica, and having one of his techs phone me to give me a half-hour primer on how to use WebEx, which seems to be some kind of enhanced internet communication method, daily messages from his secretary concerning my current state of preparation for mind-meld. So that drove me into orgies of Zhabotinsky cellular automata hacking, as that's to be, I think, the main theme of our talk.

The process is kind of science-fictional, fits in nicely with my background preparations for *Memoirs of a Crazy Mathematician*, which is what I'm leaning towards calling the novel I plan to start when *The Lifebox, the Seashell, and the Soul* is done. Or call it *Crazy Mathematicians* or possibly *Craxy Mathematicians*.

I had a flash of a three-act structure for the novel the other day. Act 1: Rutgers in New Jersey, 1972. Narrated by a guy like part of me but dumber. He has a weird friend, who's like another part of me, but smarter. The weird friend meets Gödel and learns something. I could even use my dream of Gödel's death. Act 2: The weird friend gets in a spacetime warp and pulls in the narrator. They land at the rim of Galaxy Z in the year 79982. It's the extra '2' that really bothers me. They hang out with aliens who are also mathematicians of a kind. And eventually they get themselves transmitted back to Earth. Act 3: They're computer hackers in Silicon Valley, or no, better and more transreal, Huba Kis the narrator is a CS professor at SJSU and Paul Bridge his smart friend is running a software company.

### *March 4, 2004. Wolfram's Call,* **Frek** *Arrives.*

Wolfram called on Tuesday, it was great. He had two assistants join the conference too, the four of us linked by WebEx and by MessageCenter. This meant that we could share computer desktops — like I'd be watching Wolfram's computer screen, him writing in Mathematica code and his assistants making suggestions. I had my phone on speaker mode and all four of us were on the phone as well as on the computer.

He did some quick Mathematica experiments on the Hodgepodge rule, testing if it's class 4. Mathematica has this great ability to generate textbook like illustrations in seconds.

We switched over to my desktop after awhile and then they could see my screen. I showed them some Zhabo scrolls of course. This was one spot where WebEx bogged down a little, with what they were seeing lagging a minute behind what I was seeing — Wolfram explained it was because the WebEx realtime compression algorithm was bogging down on the CAs, as these are completely different images many times a second. It was so futuristic.

Eventually Stephen started gently chiding me like he always does, about why do I look at CAs with such complicated rules as, e.g., the Double Logistic CA, with two real numbers in every cell acting as activator and inhibitor, each of them obeying Verhulst's logistic equation as well. He's a great believer in looking at the simplest possible rules, e.g., his favorites Rule 110 and Rule 30, one-dimensional CAs with a single bit per cell and with their update rules summarized in eight bits.

I never know quite how to answer him. I tried saying, "They're beautiful, like light-shows. I like finding gnarly things," but that doesn't quite go over, and doesn't really quite express what I think. Stephen he has this notion that it's more scientifically significant to find the gnarl coming from very simple things, so as to prove, that the gnarl is inherent in logic, and isn't somehow being smuggled in as scuzz stuck to the more complicated rules like bacteria dropping into a culture dish from a careless lab-worker's fingers. Stephen has an almost puritanical horror of using continuous numbers as cell states, so it was a real stretch for him to be looking at the 32-state Hodgepodge rule.

But I think it's nice to see that the same kinds of patterns arise in the complicated cases as in the simple ones. And, after all, nothing in Nature is clean and simple. All the so-called continuous computations in the world might as well be digital, with possible randomness in the final bits. But even so the world has its stable structures. They're robust against scuzz. You can get scrolls in the messiest systems, and the scrolls can be just as clean and sharp as in the simplest systems.

### *April 21, 2004. A Month of Disease*

I was sick for the last month, a tenacious bout of fever and flu leading to depression. Here's some relevant excerpts from my regular journal.

***

These days I'm mad at the book project because I feel it lured me in under false pretences. I vainly imagined I'd get a lot of money for *Lifebox* from a publisher. I could be writing a *fun* book like *Frek* and getting just as much money. A science fiction novel, or a flesh-and-blood memoir. Yeah, man, I should be writing memoirs. Yes, but I felt it was my duty to my readers to write the Tome About Computers. And who knows if I could really have turned around and sold a novel idea so fast. In all honesty, I got into doing *Lifebox* last fall because I didn't think I had the energy to start another novel. I felt drained by the two years work on *Frek* and the unexciting advance for it, and by my general hopelessness over my career.

Another beef against *Lifebox* is that it's a physically taxing job. I'm always dragging around big blocks of text and formatting illustrations. Instead of thinking and writing it often feels more like cutting and pasting — or chainsawing and welding. And I always get rushed and frenetic when I work on it. There's much to do, many loose ends, it's a hydra. I hunch my shoulders when I'm working on *Lifebox*; I tense up and rush, trying to beat down the endless blandly nodding pillow of material, I have to hit it and hit it and hit it and hit it some more, like a kid sobbing and drumming his fists on a locked door. And that's why it hurts my back. *Lifebox* up to 75,000 words now, a shade more than half done. I'm making it all up as I go along, just like my class syllabi, so that's a little stressful too. Why can't I do things the easy way? Mail it in, Ru, *mail it the hell in*!

Who cares about computers, anyway? Not the public anymore. The CS enrollment is down by a third and still falling! A nationwide trend.

***

But the book is worth doing, even though at times I think it's fatuous (I learned to use that word from Sterling who had "his" character sling it at "my" character in "Junk DNA" when we were quarrelling). What does fatuous mean, really? I'm thinking self-satisfied, superficial, stating the obvious with solemnity. As I've stewed over at length before, *Lifebox* is also skirting the dreaded zones of beating-dead-horses and looking-for-gold-coins-on-a-well-policed-parade-ground. But nobody can do it quite the way I'm doing it. Come to me, O scared Muse.

***

It also occurred to me that teaching students and writing *Lifebox* might plausibly be viewed as God's Will, in the sense that these activities make a positive contribution to society.

***

Weak and feverish. I have no thoughts of working on *Lifebox* any time soon. That's over. That's what ruined my health.

***

I'm not touching *Lifebox* for a month. Instead maybe I'll go ahead and start my novel, *Memoirs of a Crazy Mathematician*.

***

Last night I gave a talk to 120 kids aged 25-35 in San Francisco at the RX gallery. It was for a monthly event called "Dorkbot: People doing weird things with electricity." I did a little PowerPoint about the *Lifebox* title triad and about Wolfram's four classes of computation. Also read three short-short stories from *Lifebox* and showed some cellular automata demos.

***

In a way, this has all been a kind of blessing. First of all it's given me a lot more sympathy for the mentally afflicted. And, secondly, I'm currently stuck in the middle of Chapter Four of *Lifebox*, which is about the brain. And this has been a great chance to research some aspects of the brain — the mood swings, getting stuck on bad attractor. When I was at my most depressed, like Wednesday night, I got the Fall, 2004, Esalen catalog with a description I'd written of a class that Ralph Abraham and I are going to teach. And I had chirpily written, "It's calming to view moods in terms of chaos and strange attractors," which seemed so remote from the hopeless agony I was in. Hopefully my recent experience can make it more real.

Speaking of *Lifebox*, John Oakes called yesterday to tell me he's sold Four Walls Eight Windows, and my book will come out under the Thundermouth imprint of Avalon/Nation. He says he loves the first three chapters I sent him, which is nice.

My back and arm hurt so much typing this that it's still hard to imagine getting back into *Lifebox*. And I still, sigh, have the taxes to do.

I hope my life can get back to normal — what a joke. Life never does, ever, get back to normal. It's always mutating onward. The attractors continue to bifurcate, as Ralph would say.

***

My coughing finally stopped yesterday, and I'm feeling pretty good. Back into *Lifebox*.

### *June 2, 2004. Finished Chapter Four. What to say about Society?*

So now I'm done with my chapter on the mind.

In the meantime I got well, also I retired from teaching. I have till next March to finish the two remaining chapters, though I'd like to do it sooner so I can start on a novel.

I feel good about the Mind chapter, I put in almost everything I'd wanted to, and more. There's some more technical stuff that I'll do in the last chapter.

Now I face organizing something about society. It probably wouldn't hurt to reread what I've written so far, in order to smooth things. But I know if I reread I'll have to rewrite, also I'd have to take into account John Walker and Brian Silverberg's bugs, woobies, and coobies. (The last two are Walker-ese for "would be nice to change" and "could be improved.") And right now I'm more interested in pushing forward.

How did I organize the previous four chaps?

One - Computation. General idea of computing, universal machines, PCs, Net, CAs. A flow from the simple to the complex.

Two - Physics. Classical physics as parallel computation, chaos, quantum mechanics. A flow from the simple to the complex.

Three - Biology. Reproduction, morphogenesis, homeostasis, ecology, a-life, evolution. Flow from low-level to high-level processes.

Four - Mind. Reflexes, neural nets, thoughts as gliders, consciousness, personality, AI, enlightenment. Flow from low-level to high-level.

For Five - Society, I think I'll also try and flow from low to high level.

### *June 17, 2004. What's Interesting About Society?*

On the road, in the Wild West: Boulder, Colorado and Pinedale, Wyoming. Did some other stuff, now coming back to Chapter Five.

I've been reading a few pop science books about society, just now a piece of junk called *The Wisdom of Crowds*, full of self-serving horse manure about the efficacy of democracy, the free market and capitalism. It garnered a rave review in the *NY Times*, of course. These low-level pop sci books have chapters that say one thing over and over and over and over, illustrating their points with shopworn received truths and predigested news stories — without ever jumping out of the system to carry out any meta-analysis. Sheep in a cement cell, lapping at the spreading puddle of their own lukewarm urine. "This is great stuff!"

I hate so many topics that have to do with society. Fixing traffic by charging more tolls. Elections. Corporations. Committees. The sociology of science. The stock market. Sports. Triumphant movies about sports.

I think there ought to be things about society that I can love talking about, things that are real, and not just about pigs who think about power and money. How people avoid bumping into each other when they're walking on the street. How your emotions chaotically dance around when you're talking to friends and family. Bonding with people. Conversations. Status maintenance. Nabokov's remark about a lovers' conversation being like an opera aria, with the words not really mattering. A teacher in a classroom. Listening to the sounds of a city through an open window.

I'd like to break through to a radically different way of talking about society, to throw a bucket of ice-water in the face of the sleep-walking sheep who think Big Media News and The President are what matters, when these are in fact the most remote epiphenomena of society's computation in progress.

### *July 21, 2004, In Geneva for Two Weeks*

So I ended up having to come over here because of a family emergency, which kind of fizzled out. It's nice to be in Geneva, of course. And it's worked out that I've had a fair amount of time to work on the book, maybe I've even done better work than I would have if I'd stayed home. I got to spend a day discussing the book's topics with John Walker in Neuchâtel, which was useful.

I finished up section 5.1 The Human Hive and am just rounding off 5.2 Language and starting 5.3 Culture. I'm having to reach a bit, but I'm finding I have pretty much to say. Tonight I calculated that I'm better than 80% done.

The unicorn hunt is nearly ended. The baying of the hounds, the crashing of the underbrush. I smell blood.

### *August 6, 2004. Catching up on Pocket Notes*

I'm back, and it's going well. Yesterday did a great CA of the Zeldovich model for intermittency in Capow, producing a cyberspace cityscape.

***

The essence of how news spreads is the compulsion to pass it on.

(1) Impact of a cultural artifact on me. Think of the sandpile model. The artifact stimulates several nodes in my neural net. This produces a greater or lesser cascade of firings in my brain. If the effects are large, you might say that I was *primed* for this bit of news.

We might think of a brain as having several kinds of "sand," that is, different networks that can be stimulated. A gnarly message adds sand to a variety of sandpiles. An artifact that has high compression, high gnarl, may manage to stimulate a larger number of nodes in a person's neural net. Until this artifact appeared it may not have been obvious that the network was in fact primed for it. "You scratch an itch I didn't know I had."

This description to some extent fails to model the shock of a truly new idea. It's not, after all, more of the same that jacks people up. A great work of art might be viewed as creating a fresh node in the neural net, and loading the node with a large stimulation.

In this case we can again speak of the brain as having been *primed* for this input if the brain has a lot of important nodes for the new node to link to.

(2) Suppose there is a *TellThreshold*, such that if an artifact sets of a cascade with duration greater than *TellThreshold*, then I feel compelled to tell other people about the new idea. If an artifact stimulates me above the TellThreshold, then I would say that I was *primed* for this artifact.

(3) Depending upon how many people are primed for an artifact, and depending upon how many people the artifact is initially presented to, the artifact produces a greater or lesser cascade of awareness across society.

***

Arguments with your spouse obey a power law. You are in a critical state. Grudges. Two kinds of sand, red and green. Red is grudges and anger. Green is love. They overflow to the neighbors. The mind is a network, but a CA is a good approximation.

***

Do the forest fire model as well? Suppose I add a spontaneous generation term, then I think I can get BZ scrolls. Rule uses N+2 states: ready, firing, $rest_i$, with $0 <= i < n$.

(a) If firing go to $rest_0$ state.
(b) If $rest_i$ state then go to $rest_{i+1}$ state.
(c) If $rest_{N-1}$, go to ready state.
(c) If ready state and a neighbor is firing, go to firing state.
(d) If ready state, also with probability DRIVER_PROBABILIITY go to firing.
Actually, leaving out (d) this is the same as an N-LUKY rule with L=U=1, K=Y=0.
***
Society really has many colors of sand.
***
The Ramones paved the way for Green Day. Cyberpunk paved the way for Neal Stephenson. Kerouac 1956 paved the way for Kerouac 2004. Each failed artifact brings some brains closer to being primed for a derivative or reissued work.
***
I need to mention Langton's point that a city is a natural phenomenon much like a wasp hive or a copper accretion fractal.
***
A street in Manhattan, lined with human-made cars and buildings. Some men are tearing out and retrofitting some apartments. Wires run across the street bringing power. A garbage truck is picking up trash. A police car goes by. Stores sell books and summer dresses.
***
Gnarliness as a measure of art quality. Minimalism only works with an fascinating explanation. Or if you just happen to get a lucky acupuncture-type hit upon a key node. Less is less.
***
Asimov, "The Last Question," has that "now there is" line after the people as the computer if there's a god.

There's a Sheckley story about a society that asks one guy his opinions to settle an election.

"The science fictional megatext."

### *August 17, 2004. Blogging at www.boingboing.net*

It was fun, I got quite a bit of feedback, fan letters, etc., which felt nice.

I just read an expensive book about McLuhan called *On McLuhan*, with snippets from him and his followers plus the usual wasteland of cruddy big photos one expects in this kind of book. He talks a lot about, when analyzing a new medium, trying to ignore the figure and see the ground. Ignore the so-called content (which is usually repackaged older media forms) and look at how the medium changes behavior. Like cars producing roadside stands.

His friends said he made up his ideas in the context of talking, he called these mini-rants "probes."

I'm thinking about M. M. because I'd like to come up with some profound aphorism to stick into this chapter on Society and Computation in my nonfiction book, nearly done. What are the side effects of the web? I guess the blog is a side-effect. Leaving your diary out on the coffee table. What are the side effects of the blog?

The egoboo and mail bath was so nice. Like being connected. Like having a lot of friends, which I don't in meatspace. Like having a cool place to hang out, which I don't in

Los Gatos, the coffee shop though not bad, isn't what you'd call cool, filled as it is with retirees, yuppie moms, blank faced strangers, fake athletes back from fake bike rides, etc — the usual catalog of snobby putdowns ensues from this pathetic self-and-world-hating man...

I could start a fulltime blog of my own. But it was also kind of negative thing in that I had a deadline every morning. Something to live for, but isn't it nice to relax instead. What's the opposite of "the unexamined life is not worth living"? The blogged life is barely lived at all. Like the way when you have a camera, you see things differently, on the plus side you pay more attention, your eye picks out photos, on the minus side, you're dirempted, cut off, Other, observer rather than participant.

Sent this to Mark Laidlaw, he answered,

"Reminds me of a John Rechy line in, I think, *City of Night*, a put-down about a novelist: 'He had a novel where his heart should be.' The hard-core blogger has a blog there instead. I think it's a bit like everyone trying to be Harvey Pekar, but without a parade of artists coming through and adding the extra layer of artistry that makes it transcend the source material."

### *September 14, 2004. One More Section To Go*

At this point, I just wish I could finish. I got enough words, I got 147,755 of 'em.

The only section that still needs substantial work is the 6.4 section on undecidability, "Leibniz's Dream." Though I still need some more discussion of unpredictability in social computation for section 6.3. I noticed a new book by Benoit Mandelbrot about the stock market the other day, and could possibly mine this for further examples. But I'd rather not. Enuf is enuf.

I expect to top out at 150,000 words. Longer than *Bruegel*, shorter than *Frek*. Old windbag.

I got the final Answers 6.5 section done after a trip to Big Sur a couple of weeks ago. Valetudinarian tone. "Render no man evil for evil..." Pop at the end of a church service, holding up his hand, being a priest, his face shining with Godlight.

Since doing that final section, I've been getting into all this juicy unsolvability, unpredictability, and undecidability stuff for sections 6.2, 6.3, 6.4, respectively. Really nailing the meaning of the PCE and PCU. Even drew a map of "downtown" in the computational zoo today.

It's like I've set up this system of thought and now am suddenly collecting immense rewards. Climbing up the ladder I built from the trees I grew and chopped down. This morning, doing yoga, I had this deep sensual pleasure in the unsolvability, unpredictability, and undecidability of natural phenomena. The jiggles of my thoughts, the trembles of my legs. A passenger jet flying by low overhead looked as beautiful as a Rule 110 glider. I'm losing it, but in a good way.

"And through the wondering skies they came." "They" being the angels in the Xmas carol, but also the harpies of extra work, and the fractal demons of human logic-chopping.

### *September 17, 2004. Done Draft One.*

I finished *Lifebox* today. 150,000 words and about 150 illos. Posted it as a 17 Meg PDF file on a password-protected site for a few peers, and my editors. Don't plan to touch it until, say, November 1, let the comments come in and handle them all at once.

Sent the site access info to my editor and publisher, and to Stephen Wolfram, John Walker, Brian Silverman, Charles Bennett, Scott Aaronson, and my SJSU colleagues Jon Pearce, Michael Beeson, and Chris Pollett.

***September 19, 2004. Wolfram calls.***

I had an interesting talk with Stephen Wolfram today, he called, and we were on the phone for about an hour.

He's quite enthused about the book, though has only had time to leaf through it.

He had some useful suggestions we can start thinking about. He's quite the businessman, as well as the science genius.

(a) Work to remove as many of the equations/formulae as possible.

(b) Clearly visualize the target audience.

(c) Think of a subtitle that clearly explains what the book is about.

(d) Think of a concise hook for PR purposes.

(e) Have a clear plan of promotion. Hoping for a miracle doesn't work.

(f) Consider designing the book, particularly the cover, so that it looks dramatically different from all the other science books out there.

===============

(a) I'm of two minds about formulae. To me an equation makes things so much clearer. But they do scare some off. I'd kind of like to keep them. But I'll at least consider massaging out as many as I can.

(b) Wolfram's A New Kind of Science sold about 250,000 copies, so if we could simply sell to a lot of these people we'd be in good shape. Wolfram said my book in some ways made him think of Hofstadter's Gödel Escher Bach. He remarked that a book like that appeals to a bright young person who wants easy intro to complicated things. I'm also thinking a bit of the Tao of Physics crowd, as I have a certain amount about enlightenment. I think we could also reach the more professional types who like my Infinity and the Mind.

(b) This is one I'm grappling with, and it relates to (c). Ideally we get a subtitle that encapsulates a hook that's tuned to appeal to our target audience(s). Let me just brainstorm a few notions.

What Wolfram Said

A New Kind of Science for a New Kind of Century

Twenty-First Century Science

Computation and Beyond

A New Kind of Science for Everyday Life

Gnarly Computation and the Secret of Life

*Gnarly Computation, Ultimate Reality, The Meaning of Life, and How To Be Happy*

(c) I have on my desk two generic science books with exceedingly long subtitles.

*Linked: How Everything is Connected to Everything Else and What It Means for Business, Science, and Everyday Life.*

*The Wisdom of Crowds: Why the Many Are Smarter than the Few and How Collective Wisdom Shapes Business, Economics, Societies and the Nation*.

These guys have their whole "concise hook" in the subtitle! I don't particularly admire these two books, by the way, but certainly I can learn from them.

What I Learned in Silicon Valley About Life, The Universe and Everything

What I Learned in Silicon Valley About Gnarly Computation and the Secret of Life

How to Use a New Kind of Science to Change Your Life

Waking Up From the Dream of Logic To Surf the Seas of Postmodern Gnarly Computation

Secret of Reality?

Secrets of the Universe

*What one cyberpunk learned in Silicon Valley about gnarly computation, ultimate reality, the meaning of life, and how to be happy.*

(d) Wolfram mentioned of course his own *NKS* and Misner, Thorne and Wheeler's *Gravitation* as being memorably striking books-as-totem-objects. He thinks it's useful if the cover without the dust jacket look totemic as well. Should I start kicking ideas around with Georgia? I kind of like that image I made for the title page, though it would be better if it were shaped more like a brain, which could be done.

Does totemic have to mean black with yellow like NKS?

***

Oakes says stick with the equations, don't listen to Wolfie too much because, after all, W got famous first and *then* had a publishing success, he's not necessarily that knowledgeable about publishing.

Oakes likes the subtitle idea, though. I'm liking it like this:

**The Lifebox, the Seashell, and the Soul**

**What Gnarly Computation Taught Me about**
**Ultimate Reality, the Meaning Of Life, and How To Be Happy**

Or use "**Teaches Us**" instead of "**Taught Me**" to be less egotrippic? On the other hand, the transreal memoir aspect is good.

***December 3-6, 2004. Revising.***

12/3/2004. So I didn't touch the book for about two months. I was waiting for the peer suggestions, mostly I got them from Scott Aaronson and John Walker, also two of Wolfram's assistants Boguta and Crawley chimed in. There were so many nitpicks that for awhile I couldn't face them.

Meanwhile I worked on getting my novel *Mathematicians In Love* going. A nice change, getting back to fiction. Got the first chapter done, and something of a plot outline. Then I was running into plot problems so went back to face the *Lifebox* music. I did all the fixes over the last couple of weeks, by the end I was ready to choke Walker and Aaronson, which shows what a bad, ungrateful person I am. After my seventeenth day of revisions, I made the mistake of emailing them and *telling* them that I wanted to choke them; Walker's used to me and understood, I'm not sure about Scott, haven't heard back from him, better send an apology.

And now I printed out the fixed version and am rereading that and correcting. This will be my first full read-through, so I'm expecting to catch a lot of continuity things. Today I did section 1.1, what a mess, I was floundering when I wrote it, unsure where I was going and had a lot of extra crap. I cut 1,500 words from it. I hope the rest of the manuscript isn't as bad. I have to say that doing this work is hard, I'm somehow not that into the project anymore, I just want it out of my life. But maybe I'll get excited again.

I also have to check some of the photo permissions,, redo some of the drawings.

I'm suddenly feeling a little anxious about getting it all done in a reasonable time frame, as Christmas is coming up, then a trip to NYC with Sylvia, and then my planned three-week Micronesian diving trip with my brother in February. Well, we'll move it along one section at a time, see how it goes, if worst comes to worst it could take at most till the end of March.

Wolfram hasn't weighed in with any suggestions yet, he got distracted with his wife having their fourth child. I just hope he doesn't want some huge alterations at the last minute. His assistants liked the book, he says.

***

12/6/2004. So now I'm done revising Chapter One; I have a feeling that may have been the hardest one, as it's temporally the most remote from the state of mind I ended up in. I cut quite a bit, but put in some new stuff too.

I decided to drop the Chapter Summary sections, most of which weren't done, so that lightens my load a bit.

At this point the general watchword is "jettison." Prune. Simplify.

Got an email from Wolfram, he may come in with some suggestions this week after all.

Starting to feel good about the book again. It's not quite what I dreamed of producing, but it's fun to read and sporadically illuminating.

On to Chapter Two tomorrow, I hope.

### *January 17, 2005. In copy editing*

So I finished all my editing, posted the manuscript on the web, and Oakes got it and is already copyediting it.

I just visited with him in NYC. Met some of his team, the publicity people at Avalon. Since John sold Four Walls Eight Windows Avalon Publishing, my tome will come out under the Thunder's Mouth imprint.

Wolfram comments are still trickling in.

### *January 27, 2005. Marketing Suggestions From Wolfram.*

Stephen called the other day, and was on the phone for two-and-a-half hours. He was full of ideas about the book.

***

He says the ending is too "aw, shucks," or "I've written 26 books and I'm tired." I could drive things home a bit more.

***

**Short Pitch For The Lifebox, The Seashell, and the Soul**

We're presently in the midst of a third intellectual revolution. The first came with Newton: the planets obey physical laws. The second came with Darwin: biology obeys genetic laws. In today's third revolution, were coming to realize that even minds and societies emerge from interacting laws that can be regarded as computations.

Does this, then, mean that the world is dull? Far from it. The naturally occurring computations that surround us are richly complex. A tree's growth, the changes in the weather, the flow of daily news, a person's ever-changing moods --- all of these computations share the crucial property of being *gnarly*. Although lawlike and deterministic, gnarly computations are --- and this is a key point --- inherently unpredictable. The world's mystery is preserved.

Mixing together anecdotes, graphics, and fables, Rucker teases out the implications of his new worldview, which he calls "universal automatism." His analysis reveals startling aspects of the everyday world, touching upon such topics as chaos, the internet, fame, free will, and the pursuit of happiness. More than a popular science book, *The Lifebox, the Seashell, and the Soul* is a philosophical entertainment that teaches us how to enjoy our daily lives to the fullest possible extent.

### *March 31, 2005. Copy-editing.*

I'm proofreading the copy-edits of my *Lifebox* tome. I get a different copy-editor every time. I often feel that copy-editors are tendentious, ignorant, fussy, overbearing, inflexible. But maybe that's just me bridling at someone touching my sacred text. Here I'm seeing these switches: that->which, each other->one another, while->whereas, spacetime->space-time, cellular automata->CA. The replacements seem to be done somewhat mechancially. Considering matters of style or nuance, I may not want a given change every time. So I'm rolling some of them back. It feels like mud-wrestling after awhile, and I worry that I'm overlooking more important things: typos, errors, and repeated words. Well, I'll get all that on the next pass.

I did some good work today, did some fixes to the ending, put in a more NKS-thumping conclusion like Wolfram was urging me to. At this point I'm in "whatever" mode, though, and getting sloppy. Eventually I'll beef up the website with links, addenda, etc.

It's gonna be so good to be done with this. It's been two years now since I went to that NKS meeting in Boston, although really I started on the book in Sept, 2002, in Brussels, some two and a half years ago.

After months of discussion, I got a nice blurb from Wolfram! He really likes the book, which means a lot to me. He keeps saying it's better than I or my publishers realize. Here it is:

"Rudy Rucker is an outstanding prophet of what will probably be the greatest intellectual revolution of our times. This book tells the ever-surprising story of his transformation as he discovers the wonders of the computational universe, and grapples with their implications for humanity's oldest questions. For people who thrive on new ideas, this book will be a classic."

### *May 31 - June 3, 2005. Page Proofs*

I just got the page proofs of my tome, *The Lifebox, the Seashell, and the Soul*, in the mail. Some of the graphics pictures are small and blurred, the tables are quirkily formatted, the equations are oddly aligned, many of the footnote and exponent numbers are in large rather than small fonts, it's disturbing. This comes on the heels of the somewhat maddening copy-editing job.

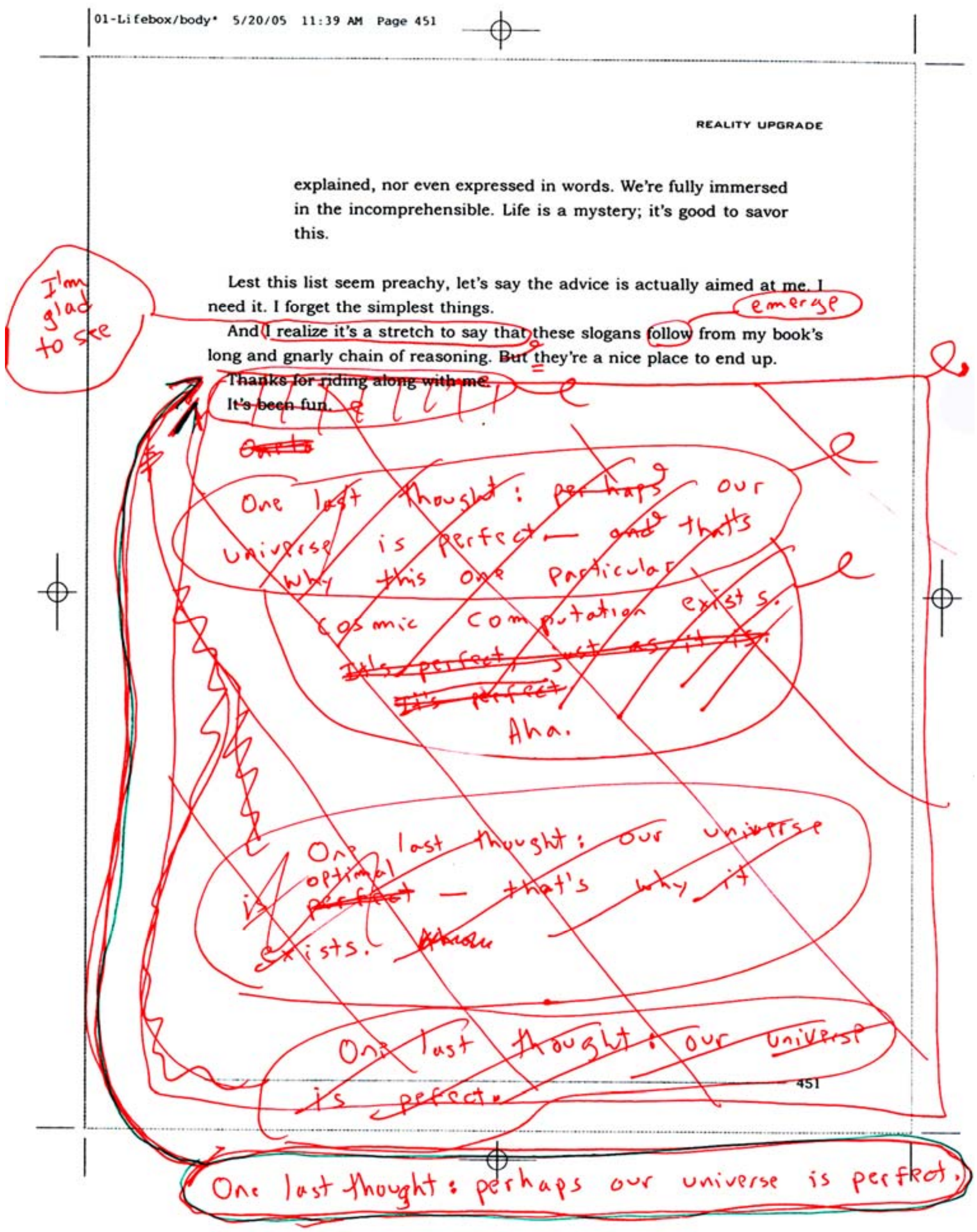
So now I gotta read through it. The I Ching throw: *Work On What Has Been Spoiled*.

On Avalon's behalf I *will* say that the made a nice spread for the book in their catalog. And they revised the cover to use a photo I took of an actual textile cone shell that Wolfram himself gave me, my personal talisman. And, of course, they were, via Oakes, one of the only houses willing to publish my idiosyncratic book.

***

So I slogged thru it and mailed it back today. Did a rewrite of the ending, made it more upbeat. Before the ending was, like, "nah, none of this true." My bad old "Imp of the Perverse" had taken over last August-September, my underminer. But I brought it back to be more hard-hitting.

Here's a picture of the revised last page.


01-Lifebox/body• 5/20/05 11:39 AM Page 451

REALITY UPGRADE

explained, nor even expressed in words. We're fully immersed in the incomprehensible. Life is a mystery; it's good to savor this.

Lest this list seem preachy, let's say the advice is actually aimed at me. I need it. I forget the simplest things.

And I realize it's a stretch to say that these slogans follow from my book's long and gnarly chain of reasoning. But they're a nice place to end up.

Thanks for riding along with me.

It's been fun.

451

I hope they'll put the changes all in, and resize the pictures I asked for. Well, now "the tome is in the mail," as Sylvia said when we went to the beach today.

***July 25, 2005. Second Round Page Proofs.***

I saw a second round of the page proofs and everything was fine. The images the right size, the footnotes fixed, the equations good, the copy-edits in place and looking reasonable after all. I polished the text a little more. I edited that cover photo of the cone shell Stephen Wolfram had given me — I noticed that in the photo you could (faintly) see the label Stephen had glued on it, "www.wolfram.com"! So got a shot of the shell without the label. Tweaked some of the drawings a bit.

So now the tome is really done, although I still have to go out and do some promo this fall. I'm happy with the book, I think it's a classic. A Magnum Opus. And I'm [probably] done with computers for good!