

# A mathematical theory of evolution and biological creativity

G. J. Chaitin, *IBM Research*

January 25, 2011

## Abstract

This essay is intended as a contribution to *The Once and Future Turing* edited by S. Barry Cooper and Andrew Hodges, which will be published by Cambridge University Press as part of the Turing Centenary celebration.

*Talk presented Monday 10 January 2011 at a workshop on “Randomness, Structure, and Causality: Measures of complexity from theory to applications” organized by Jim Crutchfield and Jon Machta at the Santa Fe Institute in New Mexico. Material displayed in boldface is what was written on the blackboard.*

I want to thank the organizers for inviting me here. I haven’t visited the Santa Fe Institute for many years. I’m delighted to be back, and I have something radically new to talk about. I think the time is now ripe to combine theoretical computer science with biology and to begin developing a theoretical mathematical biology.

## **Theoretical Biology** **Mathematical Biology**

I believe we already have the mathematical tools to begin developing such a theory.

For many years I have thought that it is a mathematical scandal that we do not have a proof that Darwinian evolution works. I want to find the simplest toy model of evolution for which I can prove that evolution will proceed forever, without ending. My emphasis is on biological creativity, something that has somehow gotten lost in the standard accounts of Darwinian evolution.

I'm aware of the fact that there is a vast literature on biology and evolution—good work, I have nothing against it—but I'm going to ignore most of it and go off in a different direction. Please bear with me.

There is a nice mathematical theory of evolution called “population genetics.” But in population genetics by definition there is no creativity, because population genetics **defines** evolution to be changes in gene frequencies in response to selective pressures, and deals with a fixed finite pool of genes. Instead, I am interested in where new genes come from, in how creativity takes place.

Another way to explain my motivation is this. The leading technology of the previous century was based on digital software, computer programming languages. And the leading technology of this new century will be bio-technology, which is based on a natural digital software, namely DNA.

***Artificial* Digital Software: Programming Languages**  
***Natural* Digital Software: DNA**

These two technologies will converge. It is no accident that people talk about computer viruses and cyber-warfare and about developing an immune system to protect cyber-assets. And what I am saying is that this isn't just a metaphor. We can take advantage of this analogy to begin developing a mathematical theory of evolution.

Darwin begins his book *On the Origin of Species* by taking advantage of the analogy between **artificial selection** by animal and plant breeders, the successful efforts of his wealthy neighbors to breed champion milk producing cows, racehorses and roses, and **natural selection** due to Malthusian limitations. I want to utilize the analogy between the random evolution of natural software, DNA, and the random evolution of artificial software, computer programs. I call this proposed new field “metabiology,” and it studies random walks in software space, hill-climbing random walks of increasing fitness.

**Evolution of Mutating Software**

## Random Walks in Software Space

Random walks are an idea that mathematicians feel comfortable with. There is a substantial literature on random walks. And I am just proposing a random walk in a richer space, the space of all possible programs in a given computer programming language, which is a space that is large enough to model biological creativity.

So I basically start with two observations. Firstly that DNA is presumably what computer scientists call a “universal programming language,” which means that it is sufficiently powerful to express **any** algorithm—in particular evo-devo teaches us to think of DNA as a computer program. Secondly, at the level of abstraction that I am working in my models, there is no essential difference between mathematical creativity and biological creativity, and so I can use mathematical problems for which there are no general methods in order to challenge my organisms and force them to keep evolving.

**DNA = universal programming language**  
**math creativity = biological creativity**

Emil Post who is forgotten but whose work was at the level of that of Kurt Gödel and Alan Turing considered that the whole point of incompleteness and uncomputability was to show the essential role of creativity in mathematics. The emphasis on formal methods provoked by the computer temporarily obliterated Post’s insight, but metabiology picks up the torch of creativity again.

To repeat, the general idea is that we are all random walks in program space! Our genomes are digital software that has been patched and modified for billions of years in order to deal with changes in the environment. In fact, I propose thinking of life as evolving software, and considering biology to be a kind of software archeology. Instead of La Mettrie’s *L’Homme machine* (1748), we now have *L’Homme software*.

To be more precise, I am studying the following toy model of evolution. I have a single organism, and I try subjecting it to random mutations. If the resulting organism is fitter, then it replaces the original organism.

Now let me explain this in more detail.

What are my organisms? Well, in his book *The Selfish Gene*, Richard Dawkins teaches us that bodies are unimportant, they are just vehicles for their genes. So I throw away the body and just keep the DNA.

A better way to explain this is to remind you of the definition of life given in John Maynard Smith and Eörs Szathmáry's books *The Major Transitions in Evolution* and *The Origins of Life*. They discuss two definitions of life. The first is fairly obvious. A living being preserves its structure while taking in matter and expelling matter, in other words it has a metabolism, and furthermore it reproduces itself. Although this seems like a natural definition, Maynard Smith and Szathmáry point out that a flame satisfies this definition.

However flames do not have heredity, fires do not remember if they were started by a match or by a cigarette lighter, and therefore they do not evolve.

Therefore Maynard Smith and Szathmáry propose a more sophisticated definition of life. **You have life when there is heredity with mutations and evolution by natural selection can take place.**

This may seem like a tautology. Darwin's theory may also seem to be a tautology—the survival of the fittest is merely the survival of the survivors—but natural selection is **not** a tautology. And this definition of life isn't either, because the whole point is to prove that there is something that satisfies the definition. The point is to find the simplest system with heredity and mutations that provably evolves.

So to make things as simple as possible, no metabolism, no bodies, only DNA. My organisms will be computer programs. I still have to explain how I do mutations, and what is my fitness measure.

For two years I worked on metabiology using what biologists call point mutations: You change/delete/add one or more contiguous bits in your computer program, and the probability of the mutations decreases exponentially with the number of bits. In this way there is a non-zero probability to go from any organism  $A$  to any other organism  $B$  in a single mutation, but if all of the bits of  $A$  are changed this probability will be infinitesimal.

With point mutations I was able to begin working, I was able to get an idea of what is going on, but the way forward was blocked; things were awkward, the way that pioneering work in math usually is. Then a few months ago, last summer, in July and August, I had a breakthrough.

I realized that from a mathematical point of view the right thing is to consider algorithmic mutations, in which a mutation is a computer program that is given the original organism  $A$  and that produces as its output the mutated organism  $B$ . If this algorithmic mutation is an  $N$ -bit program, then the mutation from  $A$  to  $B$  has probability  $2^{-N}$ .

**$N$ -bit algorithmic mutation  $A \rightarrow B$  has probability  $2^{-N}$**

If we use the prefix-free programming languages of algorithmic information theory, the total probability of all the programs will be less than one, as it should be. This is how to get what is called a “normed measure,” and this is a well-known technique.

Now there is again a non-zero probability to go from any organism  $A$  to any other organism  $B$  in a single mutation step, but the probabilities are very different. Consider, for example, the mutation that flips every bit of the program  $A$ . Before, this mutation was possible but extremely unlikely. Now it is a very simple and therefore a highly probable mutation.

If mutations are chosen at random, each mutation will be tried infinitely often, and this bit-flip mutation will be tried very frequently, a fixed proportion of the time in fact.

By the way, with algorithmic mutations it is possible that the mutation program never halts and never outputs the mutated organism  $B$ . So you cannot actually simulate our evolution model, because in order to do that you would need to use what computer scientists, following Turing, call an “oracle” for the halting problem. And we will need to use oracles again later on, to decide whether the mutated organism  $B$  is fitter than the original organism  $A$ . How do we do this? What is our fitness measure, our fitness criterion?

Well, in order to get our organisms to evolve forever, we need to challenge them with a mathematical problem that can never be solved perfectly, that can employ an arbitrary amount of mathematical creativity. Our organisms are mathematicians that are trying to become better and better, to know more and more mathematics. What math problem shall we use to force them to evolve?

The simplest extremely challenging problem is the Busy Beaver Problem, which is intimately related to Turing’s famous Halting Problem. What is the Busy Beaver Problem? That’s the problem of concisely naming an extremely large positive integer, an extremely large unsigned whole number.

Why does this require creativity? Well, suppose you have a large number  $N$  and you want to name a larger number. You can go from  $N$  to  $N + N$ , to  $N$  times  $N$ , to  $N$  to the  $N$ th power, to  $N$  raised to the  $N$ th to the  $N$ th  $N$  times. So to name large numbers you have to invent addition, multiplication, exponentiation, hyper-exponentiation, and this requires creativity.

**Busy Beaver Problem:**  $N + N, N^2, N^N, N^{N^{N^{\dots}}}$  ( $N$  times)

There is a beautiful essay on the web about this by the quantum computer complexity theorist Scott Aaronson, “Who Can Name the Biggest Number?,” which I highly recommend and that explains what a fundamental problem it is.

So that’s my fitness measure. Each of my software organisms calculates a single number, a single positive integer, and the bigger the number, the fitter the organism. The current organism  $A$  has a particular fitness  $N$ , and then we try a random mutation, according to the probability measure that I already explained, and if the resulting organism  $B$  calculates a bigger number, then it replaces  $A$ . Otherwise we try mutating  $A$  again.

Note that again we are implicitly making use of an oracle, because randomly mutating  $A$  will often produce a  $B$  that never halts and never calculates anything, so that we cannot determine if  $B$  is fitter than  $A$ —if  $B$  produces a number bigger than  $A$  does—by merely running  $B$ . We need to skip mutations that produce an invalid program  $B$ , as well as mutations that never produce a mutated organism  $B$ .

And to measure evolutionary progress, to measure the amount of biological creativity that is taking place, the rate of biological creativity, we use the so-called Busy Beaver function  $\text{BB}(N)$ , which is defined to be the biggest positive integer that can be named with a  $\leq N$  bit program. (This is a more refined version of the Busy Beaver function. The original Busy Beaver function  $\text{BB}(N)$  was the biggest integer calculated by a Turing machine with  $\leq N$  states.)

**$\text{BB}(N)$  = largest positive integer named in  $\leq N$  bits**

$\text{BB}(N)$  grows faster than any computable function of  $N$ , because  $\text{BB}(N)$  is essentially the same as the longest run-time of any  $\leq N$  bit program that eventually halts. So if  $\text{BB}(N)$  were computable that would give us a way to solve the halting problem.

Okay, now let’s see what happens if we start with a trivial organism, for example the one that calculates the number 1, and we carry out this hill-climbing random walk. We apply mutations at random and look how fast the fitness will grow. In fact, to calibrate how fast cumulative random evolution will work, let’s see where it falls between

- brainless **exhaustive search**, in which the previous organism  $A$  is ignored and we try a new organism  $B$  at random (in other words, the mutations are produced by programs that are chosen at random as before, but that are not given any input),

- and the fastest possible evolution that we can get by picking a computable sequence of mutations in the best possible manner in order to make the fitness grow quickly, which I call “**intelligent design.**”<sup>1</sup>

The designer isn’t the diety, he is the mathematician who finds the best possible sequence of mutations to try.

Here is what happens with these three different evolution regimes:

*Exhaustive Search* reaches fitness  $BB(N)$  in time  $\approx 2^N$

*Intelligent Design* reaches fitness  $BB(N)$  in time  $N$

*Cumulative Evolution* reaches fitness  $BB(N)$  in time between  $N^2$  and  $N^3$

My unit of time is the number of mutations that are tried. For example, I try one mutation per second. Note that the fitness increases faster than any computable function of the time, which shows that genuine creativity is taking place. If the organisms were improved mechanically, algorithmically, then the fitness would only grow as a computable function of the time. Of course the creativity is actually coming from the implicit use of an oracle: Each time we try a mutation and are told if the resulting  $B$  is fitter than the original organism  $A$ , we get at most one more bit of creativity and we can advance from  $BB(N)$  to  $BB(N + 1)$ . That is the best we can do, and that is what intelligent design accomplishes.

Exhaustive random search takes time that grows exponentially in  $N$  to get to  $BB(N)$ , because exhaustive search is ergodic, it is searching the entire space of possible organisms. That is not at all what happens in real evolution: The human genome has  $3 \times 10^9$  bases, but in four billion years the biosphere has only been able to try an infinitesimal fraction of the astronomical number  $4^{3 \times 10^9}$  of all possible DNA sequences of that size. Evolution is not at all ergodic.

Note that in our toy model cumulative evolution is much faster than exhaustive search, and fairly close to intelligent design. How come? In fact, what is happening in this random evolution model is that we quickly evolve very good lower bounds on the halting probability  $\Omega$ . Knowing the halting probability  $\Omega$  with  $N$  bits of precision is essentially the same as knowing  $BB(N)$ . And the random mutations  $M_K$  that rapidly increase the fitness are ones that take a lower bound on  $\Omega$  and see if they can add a 1 in the  $K$ th

---

<sup>1</sup>You cannot use an oracle to jump directly to  $BB(N)$ ,  $BB(N + 1)$ , etc., because we are only allowing a highly restricted use of oracles, in determining whether  $A \rightarrow B$  is fitter than  $A$ . Furthermore, to eliminate mutations that don’t produce a  $B$  from  $A$ .

bit position after the decimal point. In other words,  $M_K$  tries to see if the current lower bound on  $\Omega$  is still a lower bound if we add  $2^{-K}$ .

$M_K$  : **Can we add  $2^{-K}$  to our lower bound on  $\Omega$ ?**

If so, we add  $2^{-K}$  to our lower bound on  $\Omega$ . If not, we try incrementing another bit  $K$  at random. Intelligent design systematically tries  $M_K$  for  $K = 1, 2, 3, \dots$ . Cumulative evolution is not much slower, because mutation  $M_K$  essentially only needs to name  $K$ , which can be done with  $\log K + 2 \log \log K$  bits of prefix-free/self-delimiting software, and therefore has probability  $\geq 1/K(\log K)^2$  and will happen in time expected to be  $\leq K(\log K)^2$ . So cumulative evolution will try incrementing bits  $K = 1$  through  $N$  of  $\Omega$  in order by time roughly

$$\sum_{K \leq N} K(\log K)^2 \leq \text{between } N^2 \text{ and } N^3.$$

This is an outline of the proof that Darwinian evolution works in my model. For the details please see my University of Auckland Centre for Discrete Mathematics and Theoretical Computer Science Research Report CDMTCS-391 at this URL:

<http://www.cs.auckland.ac.nz/CDMTCS//researchreports/391greg.pdf>

I admit that this result seems a bit strange even to me, but I think that it **is** a first step in a metabiological direction. It is the simplest model that I can think of where you can prove that evolution works. It's my attempt to extract the mathematical essence of Darwin's theory. To my surprise, the organisms that rapidly evolve in this model are better and better lower bounds on the halting probability  $\Omega$ . In fact, the halting probabilities of all possible universal Turing machines are rapidly evolving in parallel; there are actually many halting probabilities, not just one. We will know  $N$  bits of each  $\Omega$  in time roughly between  $N$  squared and  $N$  cubed.

Why is  $\Omega$  the organism that evolves? Well, it's because a key thing in Darwin's theory is that evolution results from accumulating small changes each of which is advantageous. Darwin worried that half an eye was useless and was very concerned with the absence of intermediate forms. A chapter in his book is "On the Imperfection of the Geological Record."  $\Omega$  is jam packed with useful mathematical information and we can learn one bit of its numerical value at a time, so that better and better lower bounds on  $\Omega$  give

us a highly probable evolutionary pathway by summing small nevertheless advantageous changes.

As I said before, every mutation is tried infinitely often, and some are pretty violent. There is a mutation that replaces an organism with fitness  $N$  by a program that directly outputs  $N + 1$  without doing any computation. This is a pretty stupid organism, but it increases the fitness, and so this mutation is successful whenever it is tried. How come random evolution works in spite of such violent perturbations? Well, that's because the memory of the system resides in the fitness, which always increases. Knowing a very large integer  $N$  enables us to calculate a very good lower bound on  $\Omega$ . Just look at all the programs up to  $N$  bits in size and run them for time  $N$  to see which ones halt, and that gives you a better and better lower bound on  $\Omega$ .

So this is my current best effort to find the Platonic ideal of evolution, the simplest, most natural system that exhibits creativity and that I can prove evolves by random natural selection. We get provable evolution, which is a good first step, and which I think validates metabiology as a possible research program, but we fail to get an increase in hierarchical structure in our organisms—which are essentially just lower bounds on  $\Omega$ —and hierarchical structure is a very conspicuous feature of naturally-occurring organisms.

### What about hierarchical structure?

I actually have two more toy models of evolution that I have studied, not just the one I have explained. What varies in these models is the fitness measure, and also the programming language. In my second model I use what is called a “subrecursive” programming language, one that is not universal and for which there is no halting problem. There is no halting problem because this is a FORTRAN-like language in which each time you enter a loop you know in advance exactly how many times it is going to be executed.

And now each program calculates a function, not an integer, and the faster the function grows, the fitter the program.

$$N + N \rightarrow N^2 \rightarrow N^N \rightarrow N^{N^{N^{\dots}}} \text{ (} N \text{ times)}$$

A lot is known about subrecursive hierarchies (see for example the book by my friend Cristian Calude, *Theories of Computational Complexity*), and using all of this it is easy to show that the loop-nesting level of the programs must increase without bound. So I also have a toy model of evolution in which hierarchical structure provably emerges.

In my third toy model of evolution, the programs are once more universal, not subrecursive, and each program names what is called a “constructive Cantor ordinal number.” Here are some examples of such numbers:

$$1, 2, 3, \omega, \omega + 1, \omega + 2, 2\omega, 3\omega, \omega^2, \omega^3, \omega^\omega, \omega^{\omega^\omega}, \varepsilon_0 \dots$$

In this model I conjecture that exhaustive search is the best that you can do. In general, you expect that with arbitrary fitness landscapes exhaustive search will be needed and you will not get cumulative evolution. The fitness landscape has to be very special for Darwinian evolution to work.

So where does metabiology go from here? I expect that there is a spectrum of possible models of randomly evolving programs. More realistic models will limit the runtime of programs and thus avoid the need for oracles. I expect there to be a trade-off between biological realism and what can be proved: The more realistic the model, the more we will have to rely on computer simulations rather than proofs.

### **Are there more realistic models?**

There are many possibilities for future work. Besides limiting the run time, one can try to incorporate populations or sex. Much remains to be done. But one shouldn't expect this theoretical mathematical biology to ever become as realistic as theoretical physics. Biology is just too messy, too far removed from mathematics. And although metabiology is promising mathematically, it remains to be seen how relevant metabiology will ever be to real biology. But as my wife Virginia Chaitin points out, metabiology has already raised an interesting question for real biologists, which is how powerful are mutational mechanisms in real organisms? How closely do real organisms approach the powerful algorithmic mutations needed to make my metabiological models work?

Another caveat about metabiology is that it does not study the origin of life nor does it say anything about what may happen if we begin to take charge of our biological destiny by doing genetic engineering and producing children with designer genomes—with the best genes that money can buy.

I'd like to end with a few general remarks about biological creativity and evolution.

The conventional view is that evolution is not unceasing; you adapt perfectly to your environment, and then you stagnate. And people claim that

evolution is not about progress. The simple mathematical models that people have built up to now, in biology and in economics, talk about stability and fixed points, they do not talk about creativity.

But that is not the right way to think about biology. In biology nothing is static, everything is dynamic. Viruses, bacteria and parasites are constantly mutating, constantly probing, constantly trying to find a better way through your defenses, constantly running through all the combinatorial possibilities. Biology is ceaseless creativity, not stability, not at all. It's an arms race, and as Lewis Carroll's Red Queen said, you have to run as fast as you can to stay in the same place.<sup>2</sup>

This point is particularly well illustrated by the so-called paradox of sex that is discussed at length in the section on the rotifer in Dawkins' *The Ancestor's Tale*. In the standard view of Darwinian evolution, sex is problematic because supposedly selfish genes just want to copy themselves. But with sex you immediately throw away half of your genome, which is not at all selfish—would you call a person who gives away half of his money selfish?! Nevertheless, there are very few parthenogenetic species and sex is almost universal. How come?

### Why is there sex?

The reason is that biology is actually all about constant creativity and change; nothing is stable, just like in human affairs. And sex greatly speeds up creativity. If there are several needed mutations, sex takes the maximum of the time needed for each to occur randomly in order to get them all, whereas parthenogenetically it takes the sum of the expected mutation times instead of the maximum.

In summary, metabiology emphasizes biological creativity, not selfishness, and it opens the door to a completely new interpretation of Darwinian evolution. It remains to be seen how far this path will lead, but the first steps are encouraging. The mathematical tools are now in place to study the evolution of mutating software. Theoretical computer science **is** theoretical biology.

## Bibliography

1. S. Aaronson, <http://www.scottaaronson.com/writings/bignumbers.html>

---

<sup>2</sup>For a magnificent visual depiction of the exuberant creativity of Nature, see Ernst Haeckel's splendid books *Art Forms from the Ocean* and *Art Forms in Nature*.

2. C. Calude, *Theories of Computational Complexity*, Elsevier, 1987.
3. G. Chaitin, *Algorithmic Information Theory*, Cambridge University Press, 1987.
4. G. Chaitin, “Evolution of Mutating Software,” *EATCS Bulletin* 97 (February 2009), pp. 157–164.
5. G. Chaitin, “Metaphysics, Metamathematics and Metabiology,” *APA Newsletter on Philosophy and Computers* 10, No. 1 (Fall 2010), pp. 7–11. Also in H. Zenil, *Randomness Through Computation*, World Scientific, 2011.
6. G. Chaitin,  
<http://www.cs.auckland.ac.nz/CDMTCS//researchreports/391greg.pdf>
7. C. Darwin, *On the Origin of Species*, John Murray, 1859.
8. R. Dawkins, *The Selfish Gene*, Oxford University Press, 1976.
9. R. Dawkins, *The Ancestor’s Tale*, Houghton Mifflin, 2004.
10. E. Haeckel, *Art Forms from the Ocean*, Prestel, 2009.
11. E. Haeckel, *Art Forms in Nature*, Prestel, 2010.
12. J. O. de La Mettrie, *Man a Machine*, Open Court, 1912.
13. J. Maynard Smith, E. Szathmáry, *The Major Transitions in Evolution*, Oxford University Press, 1997.
14. J. Maynard Smith, E. Szathmáry, *The Origins of Life*, Oxford University Press, 1999.