© 1997−2004, Millennium Mathematics Project, University of Cambridge.
Permission is granted to print and copy this page on paper for non−commercial use. For other uses, including electronic redistribution, please contact us.

December 2005
Features



Omega and why maths has no TOEs

by Gregory Chaitin

Over the millennia, many mathematicians have hoped that mathematics would one day produce a Theory of Everything (TOE); a finite set of axioms and rules from which every mathematical truth could be derived. But in 1931 this hope received a serious blow: Kurt Gödel published his famous Incompleteness Theorem, which states that in every mathematical theory, no matter how extensive, there will always be statements which can't be proven to be true or false.

***Gregory Chaitin** has been fascinated by this theorem ever since he was a child, and now, in time for the centenary of Gödel's birth in 2006, he has published his own book, called Meta Math! on the subject (you can read a review in this issue of Plus). It describes his journey, which, from the work of Gödel via that of Leibniz and Turing, led him to the number Omega, which is so complex that no mathematical theory can ever describe it. In this article he explains what Omega is all about, why maths can have no Theory of Everything,*

and what this means for mathematicians.



Kurt Gödel

Splattered ink

My story begins with Leibniz in 1686, the year before Newton published his *Principia*. Due to a snow storm, Leibniz is forced to take a break in his attempts to improve the water pumps for some important German silver mines, and writes down an outline of some of his ideas, now known to us as the *Discours de métaphysique*. Leibniz then sends a summary of the major points through a mutual friend to the famous fugitive French *philosophe* Arnauld, who is so horrified at what he reads that Leibniz never sends him, nor anyone else, the entire manuscript. It languishes among Leibniz's voluminous personal papers and is only discovered and published many years after Leibniz's death.

In sections V and VI of the *Discours de métaphysique*, Leibniz discusses the crucial question of how we can distinguish a world which can be explained by science from one that cannot. How do we tell whether something we observe in the world around us is subject to some scientific law or just patternless and random? Imagine, Leibniz says, that someone has splattered a piece of paper with ink spots, determining in this manner a finite set of points on the page. Leibniz observes that, even though the points were splattered randomly, there will always be a mathematical curve that passes through this finite set of points. Indeed, many good ways to do this are now known. For example, what is called "Lagrangian interpolation" will do.

So the existence of a mathematical curve passing through a set of points cannot enable us to distinguish between points that are chosen at random and those that obey some kind of a scientific law. How, then, can we tell the difference? Well, says Leibniz, if the curve that contains the points must be extremely complex ("fort composée"), then it's not much use in explaining the pattern of the ink splashes. It doesn't really help to simplify matters and therefore isn't valid as a scientific law the points are random ("irrégulier"). The important insight here is that something is random if any description of it is extremely complex randomness is complexity.



Leibniz had a million other interests and earned a living as a consultant to princes, and as far as I know after having this idea he never returned to this subject. Indeed, he was always tossing out good ideas, but rarely, with the notable exception of the infinitesimal calculus, had the time to develop them in depth.

The next person to take up this subject, as far as I know, is Hermann Weyl in his 1932 book *The Open World*, consisting of three lectures on metaphysics that Weyl gave at Yale University. In fact, I discovered Leibniz's work on complexity and randomness by reading this little book by Weyl. And Weyl points out that Leibniz's way of distinguishing between points that are random and those that follow a law by invoking the complexity of a mathematical formula is unfortunately not too well defined: it depends on what functions you are allowed to use in writing that formula. What is complex to one person at one particular time may not appear to be complex to another person a few years later defined in this way, complexity is in the eye of the beholder.

What is complexity?

Well, the field that I invented in 1965, and which I call *algorithmic information theory*, provides a possible solution for the problem of how to measure complexity. The main idea is that any scientific law, which explains or describes mathematical objects or sets of data, can be turned into a computer program that can compute the original object or data set.



Gottfried von Leibniz

Say, for example, that you haven't splattered the ink on the page randomly, but that you've carefully placed the spots on a straight line which runs through the page, each spot exactly one centimetre away from the previous one. The theory describing your set of points would consist of four pieces of information: the equation for the straight line, the total number of spots, the precise location of the first spot, and the fact that the spots are one

centimetre apart.

You can now easily write a computer program, based on this information, which computes the precise location of each spot. In algorithmic information theory, we don't just say that such a program is based on this underlying theory, we say it *is* the theory.

This gives a way of measuring the complexity of the underlying object (in this case our ink stains): it is simply the size of the smallest computer program that can compute the object. The size of a computer program is the number of "bits" it contains: as you will know, computers store their information in strings of 0s and 1s, and each 0 or 1 is called a "bit". The more complicated the program, the longer it is and the more bits it contains. If something we observe is subject to a scientific law, then this law can be encoded as a program. What we desire from a scientific law is that it be simple the simpler it is, the better our understanding, and the more useful it is. And its simplicity or lack of it is reflected in the length of the program.

In our example, the complexity of the ink stains is precisely the length in bits of the smallest computer program which comprises our four pieces of information and can compute the location of the spots. In fact, the ink spots in this case are not very complex at all.



We have added two ideas to Leibniz's 1686 proposal. First, we measure complexity in terms of bits of information, i.e. 0s and 1s. Second, instead of mathematical equations, we use binary computer programs. Crucially, this enables us to compare the complexity of a scientific theory (the computer program) with the complexity of the data that it explains (the output of the computer program, the location of our ink stains).

As Leibniz observed, for any data there is always a complicated theory, which is a computer program that is the same size as the data. But that doesn't count. It is only a real theory if there is compression, if the program is much smaller than its output, both measured in 0/1 bits. And if there can be no proper theory, then the bit string is called *algorithmically random* or *irreducible*. That's how you define a random string in algorithmic information theory.

Let's look at our ink stains again. To know where each spot is, rather than writing down its precise location, you're much better off remembering the four pieces of information. They give a very efficient theory which explains the data. But what if you place the ink spots in a truly random fashion, by looking away and flicking your pen? Then a computer program which can compute the location of each spot for you has no choice but to store the co−ordinates that give you each location. It is just as long as its output and doesn't simplify your data set at all. In this case, there is no good theory, the data set is irreducible, or algorithmically random.

I should point out that Leibniz had the two key ideas that you need to get this modern definition of randomness, he just never made the connection. For Leibniz produced one of the first calculating machines, which he displayed at the Royal Society in London, and he was also one of the first people to appreciate base−two binary arithmetic and the fact that everything can be represented using only 0s and 1s. So, as Martin Davis argues in his book *The Universal Computer: The Road from Leibniz to Turing*, Leibniz was the first computer scientist, and he was also the first information theorist. I am sure that Leibniz would have instantly understood and appreciated the modern definition of randomness.

I should also mention that A. N. Kolmogorov also proposed this definition of randomness. He and I did this independently in 1965. Kolmogorov was at the end of his career, and I was a teenager at the beginning of my own career as a mathematician. As far as I know, neither of us was aware of the Leibniz *Discours*. But Kolmogorov never realized, as I did, that the really important application of these ideas was the new light that they shed on Gödel's incompleteness theorem and on Alan Turing's famous halting problem.

So let me tell you about that now. I'll tell you how my Omega number possesses infinite complexity and therefore cannot be explained by any finite mathematical theory. This shows that in a sense there is randomness in pure mathematics, and that there cannot be any TOE.



Omega is so complex because it's definition is based on an unsolvable problem Turing's halting problem. Let's have a look at this now.

Turing's halting problem

In 1936, Alan Turing stunned the mathematical world by presenting a model for the first digital computer, which is today known as the Turing Machine. And as soon as you start thinking about computer programs, you are faced with the following, very basic question: given any program, is there an algorithm, a sure−fire recipe, which decides whether the program will eventually stop, or whether it'll keep on running forever?

Let's look at a couple of examples. Suppose your program consists of the instruction "take every number between 1 and 10, add 2 to it and then output the result". It's obvious that this program halts after 10 steps. If, however, the instructions are "take a number x, which is not negative, and keep multiplying it by 2 until the result is bigger than 1", then the program will stop as long as the input x is not 0. If it is 0, it will keep going forever.

In these two examples it is easy to see whether the program stops or not. But what if the program is much more complicated? Of course you can simply run it and see if it stops, but how long should you wait before

you decide that it doesn't? A week, a month, a year? The basic question is whether there is a test which in a finite amount of time decides whether or not *any* given program ever halts.

And, as Turing proved, the answer is no.

What is Omega?

Now, instead of looking at individual instances of Turing's famous halting problem, you just put all possible computer programs into a bag, shake it well, pick out a program, and ask: "what is the probability that it will eventually halt?". This probability is the number Omega.

How would you like your maths displayed?

If the character

p

doesn't look like the greek letter "pi", and the character

Ö

isn't a square root sign, an alternative version of this page is available that should work with most browsers.

An example will make this clearer: suppose that in the whole wide world there are only two programs that eventually halt, and that these programs, when translated into bit strings, are 11001 and 101. Picking one of these at random is the same as randomly generating these two bit strings. You can do this by tossing a coin and writing down a 1 if heads comes up, and a 0 if tails comes up, so the probability of getting a particular bit is 1/2. This means that the probability of getting 101 is $1/2 \times 1/2 \times 1/2 = 1/2^3$, and the probability of getting 11001 is $1/2 \times 1/2 \times 1/2 \times 1/2 \times 1/2 = 1/2^5$. So the probability of randomly choosing one of these two programs is $1/2^3 + 1/2^5 = 0.15625$.



Alan Turing

Of course, in reality there are a lot more programs that halt, and Omega is the sum of lots of terms of the form $1/2^N$. Also, when defining Omega, you have to make certain restrictions on which types of programs are valid, to avoid counting things twice, and to make sure that Omega does not become infinitely large.

Anyway, once you do things properly you can define a halting probability Omega between zero and one. Omega is a perfectly decent number, defined in a mathematically rigorous way. The particular value of Omega that you get depends on your choice of computer programming language, but its surprising properties don't depend on that choice.

Why is Omega irreducible?

And what is the most surprising property of Omega? It's the fact that it is irreducible, or algorithmically random, and that it is infinitely complex. I'll try to explain why this is so: like any number we can, theoretically at least, write Omega in binary notation, as a string of 0s and 1s. In fact, Omega has an infinite binary expansion, just as the square root of two has an infinite decimal expansion

$$\sqrt{2} = 1.4142135623730950488....$$

Now the square root of two can be approximated to any desired degree of accuracy by one of many algorithms. *Newton's iteration*, for example, uses the formula

$$x_{k+1}=1/2(x_k+2/x_k)$$

to create a string of numbers which get closer and closer to the square root of two. Start with an initial value, say

$$x_1 = 1.$$

Plug x_1 into the equation to get x_2:

$$x_2 = 1/2(1+2/1) = 1.5.$$

Repeat, plugging x_2 into the equation to get x_3:

$$x_3 = 1/2(1.5+2/1.5) = 1.416666... .$$

And again:

$$x_4 = 1/2(1.41667+2/1.41667) = 1.4142156862745098039.$$

A computer program which contains this "recursive" formula and the starting value 1 can slowly chomp through the calculations and eventually compute an approximation which is as accurate as we'd like it to be. And having a good approximation means knowing a great number of the digits in the decimal expansion of the square root of two, so, given enough time, this *finite* program can compute any one of the infinitely many digits in the decimal expansion of the square root of two.

Is there a similar finite program that can compute all the bits in the binary expansion of Omega? Well, it turns out that knowing the first *N* bits of Omega gives you a way of solving the halting problem for all programs up to *N* bits in size. So, since you have a finite program that can work out all bits of Omega, you also have a finite program that can solve the halting problem for all programs, no matter what size. But this, as we know, is impossible. So such a program cannot exist.

According to our definition above, Omega is irreducible, or algorithmically random. It cannot be compressed into a smaller, finite theory. Even though Omega has a very precise mathematical definition, its infinitely many bits cannot be captured in a finite program they are just as "bad" as a string of infinitely many bits chosen at random. In fact, Omega is maximally unknowable. Even though it is precisely defined once you specify the programming language, its individual bits are maximally unknowable, maximally irreducible.

Why does maths have no TOEs?



This question is now easy to answer. A mathematical theory consists of a set of "axioms" basic facts which we perceive to be self−evident and which need no further justification and a set of rules about how to draw logical conclusions.

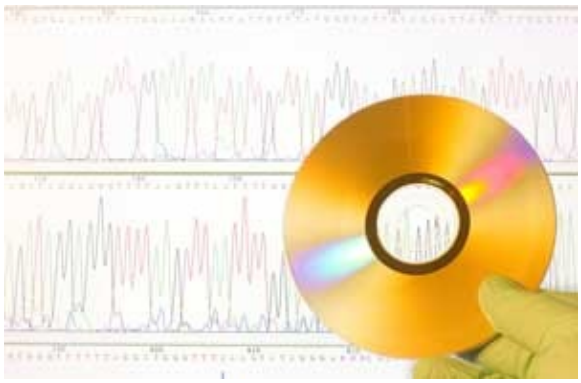
So a Theory of Everything would be a set of axioms from which we can deduce all mathematical truths and derive all mathematical objects. It would also have to have finite complexity, otherwise it wouldn't be a theory. Since it's a TOE it would have to be able to compute Omega, a perfectly decent mathematical object. The theory would have to provide us with a finite program which contains enough information to compute any one of the bits in Omega's binary expansion. But this is impossible because Omega, as we've just seen, is infinitely complex no finite program can compute it. There is no theory of finite complexity that can deduce Omega.

So this is an area in which mathematical truth has absolutely no structure, no structure that we will ever be able to appreciate in detail, only statistically. The best way of thinking about the bits of Omega is to say that each bit has probability 1/2 of being zero and probability 1/2 of being one, even though each bit is mathematically determined.

That's where Turing's halting problem has led us, to the discovery of pure randomness in a part of mathematics. I think that Turing and Leibniz would be delighted at this remarkable turn of events. Gödel's incompleteness theorem tells us that within mathematics there are statements that are unknowable, or undecidable. Omega tells us that there are in fact infinitely many such statements: whether any one of the infinitely many bits of Omega is a 0 or a 1 is something we cannot deduce from any mathematical theory. More precisely, any maths theory enables us to determine at most finitely many bits of Omega.

Where does this leave us?

Now I'd like to make a few comments about what I see as the philosophical implications of all of this. These are just my views, and they are quite controversial. For example, even though a recent critical review of two of my books in the *Notices of the American Mathematical Society* does not claim that there are any technical mistakes in my work, the reviewer strongly disagrees with my philosophical conclusions, and in fact he claims that my work has no philosophical implications whatsoever. So these are just my views, they are certainly not a community consensus, not at all.



Is maths an experimental science?

My view is that Omega is a much more disagreeable instance of mathematical incompleteness than the one found by Gödel in 1931, and that it therefore forces our hand philosophically. In what way? Well, in my opinion, in a *quasi−empirical* direction, which is a phrase coined by Imre Lakatos when he was doing philosophy in England after leaving Hungary in 1956. In my opinion, Omega suggests that even though maths and physics are different, perhaps they are not as different as most people think.

To put it bluntly, if the incompleteness phenomenon discovered by Gödel in 1931 is really serious and I believe that Turing's work and my own work suggest that incompleteness is much more serious than people think then perhaps mathematics should be pursued somewhat more in the spirit of experimental science rather than always demanding proofs for everything. Maybe, rather than attempting to prove results such as the celebrated Riemann hypothesis, mathematicians should accept that they may not be provable and simply accept them as an axiom.

At any rate, that's the way things seem to me. Perhaps by the time we reach the centenary of Turing's death in 2054, this quasi−empirical view will have made some headway, or perhaps instead these foreign ideas will be utterly rejected by the immune system of the maths community. For now they certainly are rejected. But the past fifty years have brought us many surprises, and I expect that the next fifty years will too, a great many indeed.

About this article

This article is based on one by the same author, which appeared in the August 2004 edition of Mathematics Today, on the occasion of the 50th anniversary of Alan Turing's death. This modified version appears here with the kind permission of the Institute of Mathematics and its Applications and the author.



Gregory Chaitin is at the IBM Thomas J. Watson Research Center in Yorktown Heights, New York, and is an honorary professor at the University of Buenos Aires and a visiting professor at the University of Auckland. The author of nine books, he is also a member of the International Academy of the Philosophy of Science, as well as the Honorary President of the Scientific Committee of the Institute of Complex Systems in Valparaiso, Chile. His latest book, *Meta Math!*, published by Pantheon in New York, is intended as popular science.

Download PDF version | Printer friendly version

Return to article

- contact
- copyright info
- sponsors
- privacy info

Plus is part of the family of activities in the Millennium Mathematics Project, which also includes the NRICH and MOTIVATE sites.