CDMTCS
Research
Report
Series

PARADOXES OF RANDOMNESS

Gregory Chaitin
IBM T. J. Watson Research Center,
Yorktown Heights, New York

CDMTCS-188
May 2002

Centre for Discrete Mathematics and
Theoretical Computer Science

PARADOXES OF RANDOMNESS

Gregory Chaitin[*]
chaitin@us.ibm.com
http://www.cs.auckland.ac.nz/CDMTCS/chaitin

Abstract

I'll discuss how Gödel's paradox "This statement is false/unprovable" yields his famous result on the limits of axiomatic reasoning. I'll contrast that with my work, which is based on the paradox of "The first uninteresting positive whole number," which is itself a rather interesting number, since it is precisely the first uninteresting number. This leads to my first result on the limits of axiomatic reasoning, namely that most numbers are uninteresting or random, but we can never be sure, we can never prove it, in individual cases. And these ideas culminate in my discovery that some mathematical facts are true for no reason, they are true by accident, or at random. In other words, God not only plays dice in physics, but even in pure mathematics, in logic, in the world of pure reason. Sometimes mathematical truth is completely random and has no structure or pattern that we will ever be able to understand. It is **not** the case that simple clear questions have simple clear answers, not even in the world of pure ideas, and much less so in the messy real world of everyday life.

When I was a small child I was fascinated by magic stories, because they postulate a hidden reality behind the world of everyday appearances. Later I switched to relativity, quantum mechanics, astronomy and cosmology, which also seemed quite magical and transcend everyday life. And I learned that physics says that the ultimate nature of reality is mathematical, that math is more real than the world of everyday appearances. But then I was surprised to learn of an amazing, mysterious piece of work by Kurt Gödel that pulled the rug out from under mathematical reality! How could this be?! How could Gödel show that math has limitations? How could Gödel use mathematical reasoning to show that mathematical reasoning is in trouble?!

Applying mathematical methods to study the power of mathematics is called meta-mathematics, and this field was created by David Hilbert about a century ago. He did this by proposing that math could be done using a completely artificial formal language in which you specify the rules of the game so precisely that there is a mechanical procedure to decide if a proof is correct or not. A formal axiomatic theory of the kind that Hilbert

[*] This talk was given Monday 13 May 2002 at Monash University in Melbourne, Australia. It will be available as digital video at *http://www.csse.monash.edu.au.* There are no section titles; the displayed material is what I wrote on the whiteboard as I spoke.

proposed would consist of axioms and rules of inference with an artificial grammar and would use symbolic logic to fill in all the steps, so that it becomes completely mechanical to apply the rules of inference to the axioms in every possible way and systematically deduce all the logical consequences. These are called the theorems of the formal theory.

You see, once you do this, you can forget that your formal theory has any meaning and study it from the outside as if it were a meaningless game for generating strings of symbols, the theorems. So that's how you can use mathematical methods to study the power of mathematics, if you can formulate mathematics as a formal axiomatic theory in Hilbert's sense. And Hilbert in fact thought that all of mathematics could be put into one of his formal axiomatic theories, by making explicit all the axioms or self-evident truths and all the methods of reasoning that are employed in mathematics.

In fact, Zermelo-Fraenkel set theory with the axiom of choice, ZFC, uses first-order logic and does this pretty well. And you can see some interesting work on this by my friend Jacob T. Schwartz at his website at *http://www.settheory.com.*

But then in 1931 Kurt Gödel showed that it couldn't be done, that no formal axiomatic theory could contain all of mathematical truth, that they were all incomplete. And this exploded the normal Platonic view of what math is all about.

How did Gödel do this? How can mathematics prove that mathematics has limitations? How can you use reasoning to show that reasoning has limitations?

How does Gödel show that reasoning has limits? The way he does it is he uses this paradox:

“This statement is false!”

You have a statement which says of itself that it's false. Or it says

“I'm lying!”

“I'm lying” doesn't sound too bad! But “the statement I'm making now is a lie, what I'm saying right now, this very statement, is a lie,” that sounds worse, doesn't it? This is an old paradox that actually goes back to the ancient Greeks, it's the paradox of the liar, and it's also called the Epimenides paradox, that's what you call it if you're a student of ancient Greece.

And looking at it like this, it doesn't seem something serious. I didn't take this seriously. You know, so what! Why should anybody pay any attention to this? Well, Gödel was smart, Gödel showed why this was important. And Gödel changed the paradox, and got a theorem instead of a paradox. So how did he do it? Well, what he did is he made a statement that says of itself,

“This statement is unprovable!”

Now that's a big, big difference, and it totally transforms a game with words, a situation where it's very hard to analyze what's going on. Consider

“This statement is false!”

Is it true, is it false? In either case, whatever you assume, you get into trouble, the opposite has got to be the case. Why? Because if it’s true that the statement is false, then it’s false. And if it’s false that the statement is false, then it’s true.

But with

“This statement is unprovable!”

you get a theorem out, you don’t get a paradox, you don’t get a contradiction. Why? Well, there are two possibilities. With

“This statement is false!”

you can assume it’s true, or you can assume it’s false. And in each case, it turns out that the opposite is then the case. But with

“This statement is unprovable!”

the two possibilities that you have to consider are different. The two cases are: it’s provable, it’s unprovable.

So if it’s provable, and the statement says it’s unprovable, you’ve got a problem, you’re proving something that’s false, right? So that would be very embarrassing, and you generally assume by hypothesis that this cannot be the case, because it would really be too awful if mathematics were like that. If mathematics can prove things that are false, then mathematics is in trouble, it’s a game that doesn’t work, it’s totally useless.

So let’s assume that mathematics does work. So the other possibility is that this statement

“This statement is unprovable!”

is unprovable, that’s the other alternative. Now the statement is unprovable, and the statement says of itself that it’s unprovable. Well then it’s true, because what it says corresponds to reality. And then there’s a **hole** in mathematics, mathematics is “incomplete,” because you’ve got a true statement that you can’t prove. The reason that you have this hole is because the alternative is even worse, the alternative is that you’re proving something that’s false.

The argument that I’ve just sketched is not a mathematical proof, let me hasten to say that for those of you who are mathematicians and are beginning to feel horrified that I’m doing everything so loosely. This is just the basic idea. And as you can imagine, it takes some cleverness to make a statement in mathematics that says of itself that it’s unprovable. You know, you don’t normally have pronouns in mathematics, you have to have an indirect way to make a statement refer to itself. It was a very, very clever piece of work, and this was done by Gödel in 1931.

1931

The only problem with Gödel's proof is that I didn't like it, it seemed strange to me, it seemed beside the point, I thought there had to be a better, deeper reason for incompleteness. So I came up with a different approach, another way of doing things. I found a different source for incompleteness.

Now let me tell you my approach. My approach starts off like this... I'll give you two versions, a simplified version, and a slightly less-of-a-lie version.

The simplified version is, you divide all numbers into two classes, you think of whether numbers are interesting or uninteresting, and I'm talking about whole numbers, positive integers,

$$1, 2, 3, 4, 5, \ldots$$

That's the world I'm in, and you talk about whether they're interesting or uninteresting.

Un/Interesting

Somehow you separate them into those that are interesting, and those that are uninteresting, okay? I won't tell you how. Later I'll give you more of a clue, but for now let's just keep it like that.

So, the idea is, then, if somehow you can separate all of the positive integers, the whole numbers, 1, 2, 3, 4, 5, into ones that are interesting and ones that are uninteresting, you know, each number is either interesting or uninteresting, then think about the following whole number, the following positive integer:

"The first uninteresting positive integer"

Now if you think about this number for a while, it's precisely what? You start off with 1, you ask is it interesting or not. If it's interesting, you keep going. Then you look and see if 2 is interesting or not, and precisely when you get to the first uninteresting positive integer, you stop.

But wait a second, isn't that sort of an interesting fact about this positive integer, that it's **precisely** the first uninteresting positive integer?! I mean, it stands out that way, doesn't it? It's sort of an interesting thing about it, the fact that it happens to be precisely the smallest positive integer that's uninteresting! So that begins to give you an idea that there's a problem, that there's a serious problem with this notion of interesting versus uninteresting.

Interestingly enough, last week I gave this talk at the University of Auckland in New Zealand, and Prof. Garry Tee showed me the *Penguin Dictionary of Curious and Interesting Numbers* by David Wells that was published in Great Britain in 1986. And I'll read what it says on page 120: "**39**—This appears to be the first uninteresting number, which of course makes it an especially interesting number, because it is the smallest number to have the property of being uninteresting." So I guess if you read his dictionary you will find that the entries for the positive integers 1 through 38 indicate that each of them is interesting for some reason!

And now you get into a problem with mathematical proof. Because let's assume that somehow you can use mathematics to **prove** whether a number is interesting or uninteresting. First you've got to give a rigorous definition of this concept, and later I'll explain how that goes. If you can do that, and if you can also prove whether particular positive integers are interesting or uninteresting, you get into trouble. Why? Well, just think about the first positive integer that you can **prove** is uninteresting.

"The first **provably** uninteresting positive integer"

We're in trouble, because the fact that it's precisely the first positive integer that you can **prove** is uninteresting, is a very interesting thing about it! So if there cannot be a first positive integer that you can prove is uninteresting, the conclusion is that you can **never** prove that particular positive integers are uninteresting. Because if you could do that, the first one would *ipso facto* be interesting!

But I should explain that when I talk about the first provably uninteresting positive integer I **don't** mean the smallest one, I mean the first one that you find when you systematically run through all possible proofs and generate all the theorems of your formal axiomatic theory. I should also add that when you carefully work out all the details, it turns out that you might be able to prove that a number is uninteresting, but not if its base-two representation is substantially larger than the number of bits in the program for systematically generating all the theorems of your formal axiomatic theory. So you can only prove that a finite number of positive integers are uninteresting.

So that's the general idea. But this paradox of whether you can classify whole numbers into uninteresting or interesting ones, that's just a simplified version. Hopefully it's more understandable than what I actually worked with, which is something called the Berry paradox. And what's the Berry paradox?

Berry Paradox

I showed you the paradox of the liar, "This statement is false, I'm lying, what I'm saying right now is a lie, it's false." The Berry paradox talks about

"The first positive integer that can't be named
in less than a billion words"

Or you can make it bytes, characters, whatever, you know, some unit of measure of the size of a piece of text:

Berry Paradox
"The first positive integer that can't be named
in less than a billion words/bytes/characters"

So you use texts in English to name a positive integer. And if you use texts up to a billion words in length, there are only a finite number of them, since there are only a finite number of words in English. Actually we're simplifying, English is constantly changing. But let's assume English is fixed and you don't add words and a dictionary has a finite size. So there are only a finite number of words in English, and therefore if you consider all possible texts with up to a billion words, there are a lot of them, but it's only a finite number, as mathematicians say jokingly in their in-house jargon.

And most texts in English don't name positive integers, you know, they're novels, or they're actually nonsense, gibberish. But if you go through all possible texts of up to a billion words, and there's only a finite list of them, every possible way of using an English text that size to name a number will be there somewhere. And there are only a finite number of numbers that you can name with this finite number of texts, because to name a number means to pick out one specific number, to refer to precisely one of them. But there are an infinite number of positive integers. So most positive integers, almost all of them, require more than a billion words, or any fixed number of words. So just take the first one. Since almost all of them need more than a billion words to be named, just pick the first one.

So this number is there. The only problem is, I just named it in much less than a billion words, even with all the explanation! [Laughter] Thanks for smiling and laughing! If nobody smiles or laughs, it means that I didn't explain it well! On a good day everyone laughs!

So there's a problem with this notion of naming, and this is called the Berry paradox. And if you think that the paradox of the liar, "this statement is false," or "what I'm saying now is a lie," is something that you shouldn't take too seriously, well, the Berry paradox was taken even less seriously. I took it seriously though, because the idea I extracted from it is the idea of looking at the size of computer programs, which I call program-size complexity.

Program-Size Complexity

For me the central idea of this paradox is how big a text does it take to name something. And the paradox originally talks about English, but that's much too vague! So to make this into mathematics instead of just being a joke, you have to give a rigorous definition of what language you're using and how something can name something else. So what I do is I pick a computer-programming language instead of using English or any real language, any natural language, I pick a computer-programming language instead. And then what does it mean, how do you name an integer? Well, you name an integer by giving a way to calculate it. A program names an integer if its output is that integer, you know, it outputs that integer, just one, and then it stops. So that's how you name an integer using a program.

And then what about looking at the size of a text measured in billions of words? Well, you don't want to talk about words, that's not a convenient measure of software size. People in fact in practice use megabytes of code, but since I'm a theoretician I use bits. You know, it's just a multiplicative constant conversion factor! In biology the unit is kilobases, right? So every field has its way of measuring information.

Okay, so what does it mean then for a number to be interesting or uninteresting, now that I'm giving you a better idea of what I'm talking about. Well, interesting means it stands out some way from the herd, and uninteresting means it can't be distinguished really, it's sort of an average, typical number, one that isn't worth a second glance. So how do you define that mathematically using this notion of the size of computer programs? Well, it's very simple: a number is uninteresting or algorithmically random or irreducible or incompressible if there's no way to name it that's more concise than just writing out the number directly. That's the idea.

In other words, if the most concise computer program for calculating a number just says to print 123796402, in that case, if that's the best you can do, then that number is uninteresting. And that's typically what happens. On the other hand, if there is a small, concise computer program that calculates the number, that's atypical, that means that it has some quality or characteristic that enables you to pick it out and to compress it into a smaller algorithmic description. So that's unusual, that's an interesting number.

Once you set up this theory properly, it turns out that most numbers, the great majority of positive integers, are uninteresting. You can prove that as a theorem. It's not a hard theorem, it's a counting argument. There can't be a lot of interesting numbers, because there aren't enough concise programs. You know, there are a lot of positive integers, and if you look at programs with the same size in bits, there are only about as many programs of the same size as there are integers, and if the programs have to be smaller, then there just aren't enough of them to name all of those different positive integers.

So it's very easy to show that **the vast majority** of positive integers cannot be named substantially more concisely than by just exhibiting them directly. Then my key result becomes, that in fact you can never prove it, not in **individual** cases! Even though most positive integers are uninteresting in this precise mathematical sense, you can never be sure, you can never prove it—although there may be a finite number of exceptions. But you can only prove it in a small number of cases. So most positive integers are uninteresting or algorithmically incompressible, but you can almost never be sure in individual cases, even though it's overwhelmingly likely.

That's the kind of "incompleteness result" I get. (That's what you call a result stating that you can't prove something that's true.) And my incompleteness result has a very different flavor than Gödel's incompleteness result, and it leads in a totally different direction. Fortunately for me, everyone liked the liar paradox, but nobody took the Berry paradox really seriously!

Let me give you another version of this result. Let's pick a computer programming language, and I'll say that a computer program is **elegant** if no program that is smaller than it is produces the same output that it does. Then you can't prove that a program is elegant if it's substantially larger than the algorithm for generating all the theorems of the formal axiomatic system that you are using, if that's written in that same computer programming language. Why?

Well, start generating all the theorems until you find the first one that proves that a particular computer program that is larger than that is elegant. That is, find the first provably elegant program that's larger than the program in the same language for generating all the theorems. Then run it, and its output will be your output.

I've just described a program that produces the same output as a provably elegant program, but that's smaller than it is, which is impossible! This contradiction shows that you can only prove that a finite number of programs are elegant, if you are using a fixed formal axiomatic theory.

By the way, this implies that you can't always prove whether or not a program halts, because if you could do that then it would be easy to determine whether or not a program is elegant. So I'm really giving you an information-theoretic perspective on what's called Turing's halting problem, I'm connecting that with the idea of algorithmic information and with program-size complexity.

I published an article about all of this in *Scientific American* in 1975, it was called "Randomness and mathematical proof," and just before that I called Gödel on the phone to tell him about it, that was in 1974.

I was working for IBM in Buenos Aires at the time, and I was visiting the IBM Watson Research Center in New York—that was before I joined IBM Research permanently. And just before I had to go back to Buenos Aires I called Gödel on the phone at the Princeton Institute for Advanced Study and I said, "I'm fascinated by your work on incompleteness, and I have a different approach, using the Berry paradox instead of the paradox of the liar, and I'd really like to meet you and tell you about it and get your reaction." And he said, "It doesn't make any difference which paradox you use!" (And his 1931 paper said that too.) I answered, "Yes, but this suggests to me a new information-theoretic view of incompleteness that I'd very much like to tell you about." He said, "Well, send me a paper on this subject and call me back, and I'll see if I give you an appointment."

I had one of my first papers then, actually it was the proofs of one of my first papers on the subject. It was my 1974 *IEEE Information Theory Transactions* paper; it's reprinted in Tymoczko, *New Directions in the Philosophy of Mathematics.* And I mailed it to Gödel. And I called back. And incredibly enough, he made a small technical remark, and he gave me an appointment. I was delighted, you can imagine, my hero, Kurt Gödel! And the great day arrives, and I'm in my office in the Watson Research Center at Yorktown Heights, NY, and it was April 1974, spring. In fact, it was the week before Easter. And I didn't have a car. I was coming from Buenos Aires, I was staying at the YMCA in White Plains, but I figured out how to get to Princeton, New Jersey by train. You know, I'd take the train into New York City and then out to Princeton. It would only take me three hours, probably, to do it!

So I'm in my office, ready to go, almost, and the phone rings. And I forgot to tell you, even though it was the week before Easter, it had snowed. It wasn't a whole lot of snow; you know, nothing would stop me from visiting my hero Gödel at Princeton. So anyway, the phone rings, and it's Gödel's secretary, and she says, "Prof. Gödel is extremely careful about his health, and because it's snowed, he's not going to be coming in to the Institute today, so your appointment is canceled!"

And as it happened, that was just two days before I had to take a plane back to Buenos Aires from New York. So I didn't get to meet Gödel! This is one of the stories that I put in my book *Conversations with a Mathematician.*

So all it takes is a new idea! And the new idea was waiting there for anybody to grab it. The other thing you have to do when you have a new idea is, don't give up too soon. As George Polya put it in his book *How to Solve It,* theorems are like mushrooms, usually where there's one, others will pop up! In other words, another way to put it, is that usually the difference between a professional, expert mathematician with lots of experience and a young, neophyte mathematician is not that the older mathematician has more ideas. In fact, the opposite is usually the case. It's usually the kids that have all the fresh ideas! It's that the professional knows how to take more advantage of the ideas he has. And one of the things you do, is you don't give up on an idea until you get all the milk, all the juice out of it!

So what I'm trying to lead up to is that even though I had an article in *Scientific American* in 1975 about the result I just told you, that most numbers are random, algorithmically random, but you can never prove it, I didn't give up, I kept thinking about it. And sure enough, it turned out that there was another major result there, that I described in my article in *Scientific American* in 1988. Let me try to give you the general idea.

The conclusion is that

**Some mathematical facts
are true for no reason,
they're true by accident!**

Let me just explain what this means, and then I'll try to give an idea of how I arrived at this surprising conclusion. The normal idea of mathematics is that if something is true it's true for a reason, right? The reason something is true is called a proof. And a simple version of what mathematicians do for a living is they find proofs, they find the reason that something is true.

Okay, what I was able to find, or construct, is a funny area of pure mathematics where things are true for no reason, they're true by accident. And that's why you can never find out what's going on, you can never prove what's going on. More precisely, what I found in pure mathematics is a way to model or imitate, independent tosses of a fair coin. It's a place where God plays dice with mathematical truth. It consists of mathematical facts which are so delicately balanced between being true or false that we're never going to know, and so you might as well toss a coin. You can't do better than tossing a coin. Which means the chance is half you're going to get it right if you toss the coin and half you'll get it wrong, and you can't really do better than that.

So how do I find this complete lack of structure in an area of pure mathematics? Let me try to give you a quick summary. For those of you who may have heard about it, this is what I like to call Ω, it's a real number, the halting probability.

Omega Number
"Halting Probability"

And some people are nice enough to call this "Chaitin's number." I call it Ω. So let me try to give you an idea of how you get to this number. By the way, to show you how much interest there is in Ω, let me mention that this month there is a very nice article on Ω numbers by Jean-Paul Delahaye in the French popular science magazine *Pour la Science,* it's in the May 2002 issue.

Well, following Vladimir Tasić, *Mathematics and the Roots of Postmodern Thought,* the way you explain how to get to this number that shows that some mathematical facts are true for no reason, they're only true by accident, is you start with an idea published by Émile Borel in 1927, of using one real number to answer all possible yes/no questions, not just mathematical questions, all possible yes/no questions in English—and in Borel's case it was questions in French. How do you do it?

Well, the idea is you write a list of all possible questions. You make a list of all possible questions, in English, or in French. A first, a second, a third, a fourth, a fifth:

Question # 1
Question # 2
Question # 3
Question # 4
Question # 5

The general idea is you order questions say by size, and within questions of the same size, in some arbitrary alphabetical order. You number all possible questions.

And then you define a real number, Borel's number, it's defined like this:

Borel's Number

$.d_1d_2d_3d_4d_5$

The Nth digit after the decimal point, d_N,
answers the Nth question!!

Well, you may say, most of these questions are going to be garbage probably, if you take all possible texts from the English alphabet, or French alphabet. Yes, but a digit has ten possibilities, so you can let 1 mean the answer is true, 2 mean the answer is false, and 3 mean it's not a valid yes/no question in English, because it's not valid English, or it is valid English, but it's not a question, or it is a valid question, but it's not a yes/no question, for example, it asks for your opinion. There are various ways to deal with all of this.

So you can do all this with one real number—and a real number is a number that's measured with infinite precision, with an infinite number of digits d_N after the decimal point—you can give the answers to all yes/no questions! And these will be questions about history, questions about philosophy, questions about mathematics, questions about physics.

It can do this because there's an awful lot you can put into a real number. It has an infinite amount of information, because it has an infinite number of digits. So this is a way to say that real numbers are very **un**real, right? So let's start with this very unreal number that answers all yes/no questions, and I'll get to my Ω number in a few steps.

The next step is to make it only answer questions about Turing's halting problem. So what's Turing's halting problem? Well, the halting problem is a famous question that Turing considered in 1936. It's about as famous as Gödel's 1931 work, but it's different.

Turing's Halting Problem 1936
[1931 Gödel]

And what Turing showed is that there are limits to mathematical reasoning, but he did it very differently from Gödel, he found something concrete. He doesn't say "this statement is unprovable" like Gödel, he found something concrete that mathematical reasoning can't do: it can't settle in advance whether a computer program will ever halt. This is the halting problem, and it's in a wonderful paper, it's the beginning of theoretical computer science, and it was done before there were computers. And this is the Turing who then went on and did important things in cryptography during the Second World War, and built computers after the war. Turing was a Jack of all trades.

So how do you prove Turing's result that there's no algorithm to decide if a computer program—a self-contained computer program—will ever halt? (Actually the problem is to decide that it will **never** halt.) Well, that's not hard to do, in many different ways, and I sketched a proof before, when I was talking about proving that programs are elegant.

So let's take Borel's real number, and let's change it so that it only answers instances of the halting problem. So you just find a way of numbering all possible computer programs, you pick some fixed language, and you number all programs somehow: first program, second program, third program, you make a list of all possible computer programs in your mind, it's a mental fantasy.

Computer Program # 1
Computer Program # 2
Computer Program # 3
Computer Program # 4
Computer Program # 5

And then what you do is you define a real number whose Nth digit—well, let's make it binary now instead of decimal—whose Nth bit tells us if the Nth computer program ever halts.

Turing's Number

$.b_1b_2b_3b_4b_5$
The Nth bit after the binary point, b_N,
tells us if the Nth computer program ever halts.

So we've already economized a little, we've gone from a decimal number to a binary number. This number is between zero and one, and so is Borel's number, there's no integer part to this real number. It's all in the fractional part. You have an infinite number of digits or bits after the decimal point or the binary point. In the previous number, Borel's original one, the Nth digit answers the Nth yes/no question in French. And here the Nth bit of this new number, Turing's number, will be 0 if the Nth computer program never halts, and it'll be 1 if the Nth computer program does eventually halt.

So this one number would answer all instances of Turing's halting problem. And this number is uncomputable, Turing showed that in his 1936 paper. There's no way to calculate this number, it's an uncomputable real number, because the halting problem is unsolvable. This is shown by Turing in his paper.

So what's the next step? This still doesn't quite get you to randomness. This number gets you to uncomputability. But it turns out this number, Turing's number, is redundant. Why is it redundant?

Redundant

Well, the answer is that there's a lot of repeated information in the bits of this number. We can actually compress it more, we don't have complete randomness yet. Why is there a lot of redundancy? Why is there a lot of repeated information in the bits of this number? Well, because different cases of the halting problem are connected. These bits b_N are not independent of each other. Why?

Well, let's say you have K instances of the halting problem. That is to say, somebody gives you K computer programs and asks you to determine in each case, does it halt or not.

$$K \text{ instances of the halting problem?}$$

Is this K bits of mathematical information? K instances of the halting problem will give us K bits of Turing's number. Are these K bits independent pieces of information? Well, the answer is no, they never are. Why not? Because you don't really need to know K yes/no answers, it's not really K full bits of information. There's a lot less information. It can be compressed. Why?

Well, the answer is very simple. If you have to ask God or an oracle that answers yes/no questions, you don't really need to ask K questions to the oracle, you don't need to bother God that much! You really only need to know what? Well, it's sufficient to know **how many of the programs halt.**

And this is going to be a number between zero and K, a number that's between zero and K.

$$0 \leq \# \text{ that halt} \leq K$$

And if you write this number in binary it's really only about $\log_2 K$ bits.

$$\# \text{ that halt} = \log_2 K \text{ bits}$$

If you know how many of these K programs halt, then what you do is you just start running them all in parallel until you find that precisely that number of programs have halted, and at that point you can stop, because you know the other ones will never halt. And knowing how many of them halt is a lot less than K bits of information, it's really only about $\log_2 K$ bits, it's the number of bits you need to be able to express a number between zero and K in binary, you see.

So different instances of the halting problem are never independent, there's a lot of redundant information, and Turing's number has a lot of redundancy. But essentially just by using this idea of telling how many of them halt, you can squeeze out all the redundancy. You know, the way to get to randomness is to remove redundancy! You distill it, you concentrate it, you crystallize it. So what you do is essentially you just take advantage of this observation—it's a little more sophisticated than that—and what you get is my halting probability.

So let me write down an expression for it. It's defined like this:

Omega Number

$$\Omega = \textstyle\sum_{p \text{ halts}} 2^{-|p|}$$

$$|p| = \text{size in bits of program } p$$

$$0 < \Omega < 1$$

Then write Ω in binary!

So this is how you get randomness, this is how you show that there are facts that are true for no reason in pure math. You define this number Ω, and to explain this I would take a long time and I don't have it, so this is just a tease!

For more information you can go to my books. I actually have **four** small books published by Springer-Verlag on this subject: *The Limits of Mathematics, The Unknowable, Exploring Randomness* and *Conversations with a Mathematician.* These books come with LISP software and a Java applet LISP interpreter that you can get at my website.

So you define this Ω number to be what? You pick a computer programming language, and you look at all programs p that halt, p is a program, and you sum over all programs p that halt. And what do you sum? Well, if the program p is K bits long, it contributes $1/2^K$, one over two to the K, to this halting probability.

In other words, each K-bit program has probability $1/2^K$, and you'll immediately notice that there are two to the thousand thousand-bit programs, so probably this sum will diverge and give infinity, if you're not careful. And the answer is yes, you're right if you worry about that. So you have to be careful to do things right, and the basic idea is that no extension of a valid program is a valid program. And if you stipulate that the programming language is like that, that its programs are "self-delimiting," then this sum is in fact between zero and one and everything works. Okay?

Anyway, I don't want to go into the details because I don't have time. So if you do everything right, this sum

$$\sum_{p \text{ halts}} 2^{-|p|}$$

actually converges to a number between zero and one which is the halting probability Ω. This is the probability that a program, each bit of which is generated by an independent toss of a fair coin, eventually halts. And it's a way of summarizing all instances of the halting problem in one real number and doing it so cleverly that there's no redundancy.

So if you take this number and then you write it in binary, this halting probability, it turns out that those bits of this number written in binary, these are independent, irreducible mathematical facts, there's absolutely no structure. Even though there's a simple mathematical definition of Ω, those bits, if you could see them, could not be distinguished from independent tosses of a fair coin. There is no mathematical structure that you would ever be able to detect with a computer, there's no algorithmic pattern, there's no structure that you can capture with mathematical proofs—even though Ω has a simple mathematical definition. It's incompressible, irreducible mathematical information. And the reason is, because if you knew the first N bits of this number Ω, it would solve the halting problem for all programs up to N bits in size, it would enable you to answer the halting problem for all programs p up to N bits in size. That's how you prove that this Ω number is random in the sense I explained before of being algorithmically incompressible information.

And that means that not only you can't compress it into a smaller algorithm, you can't compress it into fewer bits of axioms. So if you wanted to be able to determine K bits of Ω, you'd need K bits of axioms to be able to prove what K bits of this number are. It has—its bits have—no structure or pattern that we are capable of seeing.

However, you **can** prove all kinds of nice mathematical theorems about this Ω number. Even though it's a specific real number, it really mimics independent tosses of a fair coin. So for example you can prove that 0's and 1's happen in the limit exactly fifty percent of the time, each of them. You can prove all kinds of **statistical** properties, but you can't determine **individual** bits!

So this is the strongest version I can come up with of an incompleteness result...

Actually, in spite of this, Cristian Calude, Michael Dinneen and Chi-Kou Shu at the University of Auckland have just succeeded in calculating the first 64 bits of a particular Ω number. The halting probability Ω actually depends on the choice of computer or programming language that you write programs in, and they picked a fairly natural one, and were able to decide which programs less than 85 bits in size halt, and from this to get the first 64 bits of this particular halting probability.

This work by Calude *et alia* is reported on page 27 of the 6 April 2002 issue of the British science weekly *New Scientist,* and it's also described in Delahaye's article in the May 2002 issue of the French monthly *Pour la Science,* and it'll be included in the second edition of Calude's book on *Information and Randomness,* which will be out later this year.

But this doesn't contradict my results, because all I actually show is that an N-bit formal axiomatic system can't enable you to determine substantially more than N bits of the halting probability. And by N-bit axiomatic system I mean one for which there is an N-bit program for running through all possible proofs and generating all the theorems. So you might in fact be able to get some initial bits of Ω.

Now, what would Hilbert, Gödel and Turing think about all of this?!

I don't know, but I'll tell you what I think it means, it means that math is different from physics, but it's not that different. This is called the quasi-empirical view of mathematics, and Tymoczko has collected a bunch of interesting papers on this subject, in his book on *New Directions in the Philosophy of Mathematics.* This is also connected with what's called experimental mathematics, a leading proponent of which is Jonathan Borwein, and there's a book announced called *The Experimental Mathematician* by Bailey, Borwein and Devlin that's going to be about this. The general idea is that proofs are fine, but if you can't find a proof, computational evidence can be useful, too.

Now I'd like to tell you about some questions that I don't know how to answer, but that I think are connected with this stuff that I've been talking about. So let me mention some questions I **don't** know how to answer. They're not easy questions.

Well, one question is positive results on mathematics:

Positive Results

Where do new mathematical concepts come from?

I mean, Gödel's work, Turing's work and my work are negative in a way, they're incompleteness results, but on the other hand, they're positive, because in each case you introduce a new concept: incompleteness, uncomputability and algorithmic randomness. So in a sense they're examples that mathematics goes forward by introducing new concepts! So how about an optimistic theory instead of negative results about the limits of mathematical reasoning? In fact, these negative metamathematical results are taking place in a century which is a tremendous, spectacular success for mathematics, mathematics is advancing by leaps and bounds. So there's no reason for pessimism. So what we need is a more realistic theory that gives us a better idea of **why** mathematics is doing so splendidly, which it is. But I'd like to have some theoretical understanding of this, not just anecdotal evidence, like the book about the Wiles proof of Fermat's result.[1]

So this is one thing that I don't know how to do and I hope somebody will do.

Another thing which I think is connected, isn't where new mathematical ideas come from, it's where do new biological organisms come from. I want a theory of evolution, biological evolution.[2]

Biological Evolution

Where do new biological ideas come from?

You see, in a way biological organisms are ideas, or genes are ideas. And good ideas get reused. You know, it's programming, in a way, biology.

Another question isn't theoretical evolutionary biology—which doesn't exist, but that is what I'd like to see—another question is where do new ideas come from, not just in math! **Our** new ideas. How does the brain work? How does the mind work? Where do new ideas come from? So to answer that, you need to solve the problem of AI or how the brain works!

AI/Brain/Mind

Where do new ideas come from?

[1] Simon Singh, *Fermat's Enigma;* see also the musical *Fermat's Last Tango.*

[2] In Chapter 12 of *A New Kind of Science,* Stephen Wolfram says that he thinks there is nothing to it, that you get life right away, we're just universal Turing machines, but I think there's more to it than that.

In a sense, where new mathematical concepts come from is related to this, and so is the question of the origin of new biological ideas, new genes, new ideas for building organisms—and the ideas keep getting reused. That's how biology seems to work. Nature is a cobbler!—So I think these problems are connected, and I hope they have something to do with the ideas I mentioned, my ideas, but perhaps not in the form that I've presented them here.

So I don't know how to answer these questions, but maybe some of you will be able to answer them. I hope so! The future is yours, do great things!

Thank you!

References

1. D. Bailey, J. Borwein, K. Devlin, *The Experimental Mathematician,* A. K. Peters, to appear.

2. C. Calude, *Information and Randomness,* Springer-Verlag, 2002.

3. G. J. Chaitin, "Information-theoretic computational complexity," *IEEE Information Theory Transactions,* 1974, pp. 10–15.

4. G. J. Chaitin, "Randomness and mathematical proof," "Randomness in arithmetic," *Scientific American,* May 1975, July 1988, pp. 47–52, 80–85.

5. G. J. Chaitin, *The Limits of Mathematics, The Unknowable, Exploring Randomness, Conversations with a Mathematician,* Springer-Verlag, 1998, 1999, 2001, 2002.

6. M. Chown, "Smash and grab," *New Scientist,* 6 April 2002, pp. 24–28.

7. J.-P. Delahaye, "Les nombres oméga," *Pour la Science,* May 2002, pp. 98–103.

8. G. Polya, *How to Solve It,* Princeton University Press, 1988.

9. J. Rosenblum, J. S. Lessner, *Fermat's Last Tango,* Original Cast Records OC-6010, 2001.

10. S. Singh, *Fermat's Enigma,* Walker and Co., 1997.

11. V. Tasić, *Mathematics and the Roots of Postmodern Thought,* Oxford University Press, 2001.

12. T. Tymoczko, *New Directions in the Philosophy of Mathematics,* Princeton University Press, 1998.

13. D. Wells, *The Penguin Dictionary of Curious and Interesting Numbers,* Penguin Books, 1986.

14. S. Wolfram, *A New Kind of Science,* Wolfram Media, 2002.