CHAITIN ARTICLES

D. Gligoroski, A. Dimovski
Institute of Informatics, Faculty of Natural Sciences and Mathematics,
Sts. Cyril and Methodius University,
Arhimedova bb, PO BOX 162, Skopje, Macedonia
{jasen, infolab, marjan}@ii.edu.mk
gligoroski@yahoo.com; alexd7mk@yahoo.com

Abstract: In this paper, we'll discuss how to Godel's paradox "This statement is false / unprovable" which yields his famous result on the limits of axiomatic reasoning, Chaitin contrasts that with his theory, which is based on the paradox of "The first uninteresting positive whole number". This paradox leads to results on the limits of axiomatic reasoning, namely the most part of numbers are uninteresting or random, but we can never be sure, we can never prove it, in individual cases. And these ideas culminate in 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.

Keywords: Godel's theory, Berry paradox, program-size complexity, Borel's number, Turing's halting problem, Turing's number, redundant, Omega number, positive results

1. Introduction

Here, we'll talk about stuff that looks impractical, because it's basically philosophy, epistemology, and limits of knowledge, but one of Chaitin's basic theorems in this field, and maybe you could even say that it's the single most important result in the field, is that the field is useless for practical applications! We'll explain this later.

Starting with Kurt Godel's work, you sure ask yourself how could mathematics prove that mathematics has limitations. How could you use reasoning to show that reasoning has limitations? It was a good idea for Chaitin, to build his own theory on the limits of reasoning. Anyway, let's tell how Godel did it, to set the stage a little bit, so then we can compare and contrast the way Chaitin did it.

2. Godel's theory

So let's see how Godel shows that reasoning has limits. And the way he does it is he uses this paradox:

"This statement is false!"

So there's a statement that says of itself that it's false. Or it says

"I'm lying!"

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.

And looking at it like this, it doesn't seem something serious. But, Godel was smart, Godel showed why this was important. He changed the paradox, and got a theorem instead of a paradox. So how did he do it? Well, he made a statement that says

"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, because 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. 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 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; this is just the basic idea. It takes some cleverness to make a statement in mathematics that says of itself that it's unprovable. It was a very, very clever piece of work, and Godel did this in 1931.

3. Uninteresting and interesting numbers

So that's the way Godel did it. But Chaitin had own, much different approach. He takes whole positive integers,

0, 1, 2, 3, 4, 5, . . .

And question is whether they're interesting or uninteresting. Somehow you can separate them into those that are interesting, and those that are uninteresting. So, the idea is, if somehow you can separate all of the positive integers, into ones that are interesting and ones that are uninteresting, so each number is either interesting or uninteresting, then think about the following whole positive integer:

"The first uninteresting positive integer"

You start off with 0, you ask is it interesting or not. If it's interesting, you keep going. Then you look and see if 1 is interesting or not, and precisely when you get to the first uninteresting positive integer, you stop.

But the question is, isn't that sort of an interesting fact about this positive integer, that it's precisely the first uninteresting positive integer. 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.

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. If you can do that, and if you can also prove whether particular positive integers are interesting or uninteresting, you get into trouble. 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 be interesting!

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. But what Chaitin actually worked with, which is something called the Berry paradox.

4. Berry Paradox

The Berry paradox is

"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.

And most texts in English don't name positive integers, 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. 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.

The only problem is, I just named it in much less than a billion words, even with all the explanation.

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 is something that you shouldn't take too seriously, well, the Berry paradox was taken even less seriously. But Chaitin took it seriously though, because the idea he extracted from it is the idea of looking at the size of computer programs, which is called program-size complexity.

5. Program-Size Complexity

The central idea of Berry 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 we pick a computer-programming language instead of using English or any real language, any natural language. Now how do you name an integer? You name an integer by giving a way to calculate it. A program names an integer if its output is 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? You don't want to talk about words, that's not a convenient measure of software size. So we'll use bits.

Let's explain better, what does it mean then for a number to be interesting or uninteresting. 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.

In other words, if the most concise computer program for calculating a number is huge, and if that's the best you can do, then that number is uninteresting. 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. So it's very easy to show that the vast majority of positive integers cannot be named more concisely than by just exhibiting them directly. Then the 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, not in individual cases.

So most positive integers are uninteresting or algorithmically incompressible, but you can almost never be sure in individual cases. That's the kind of "incompleteness result". (That's what you call a result stating that you can't prove something that's true.) And this incompleteness result has a very different flavor than Godel's incompleteness result, and it leads in a totally different direction.

The conclusion is that:

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

Let's just explain what this means, and then we'll try to give an idea of how Chaitin arrived at this surprising conclusion. The normal idea of mathematics is that if something is true it's true for a reason. 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.

Let's try to find, or construct, an 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, how can we find a way in pure mathematics 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 beings true or false that we're never going to know.

So how did Chaitin find this complete lack of structure in an area of pure mathematics? Let's try to give you a quick summary.

6. Omega Number - "Halting Probability"

Omega number is also called "Chaitin's number", and here we'll try to give you an idea of how you get to this number.

The way you start getting 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 that goes back to Borel almost a century ago, 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.

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

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

: d1d2d3d4d5

The Nth digit after the decimal point, dN;

answers the Nth question!

A digit from Borel’s number has ten possibilities, so 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. This number has an infinite amount of information, because it has an infinite number of digits.

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

7. Turing's Halting Problem 1936

Turing showed that there are limits to mathematical reasoning, but he did it very differently from Godel, 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 the beginning of theoretical computer science, and it was done before there were computers.

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

Computer Program # 1

Computer Program # 2

Computer Program # 3

Computer Program # 4

And then what you do is you define a binary number whose Nth bit tell us if the Nth computer program ever halts.

Turing's Number

:b1b2b3b4b5

The Nth bit after the binary point, bN;

tells us if the Nth computer program ever halts.

In the Borel's original number, 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 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?

The answer is that there's a lot of repeated information in the bits of this number. We can actually compress it more.

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 instances of the halting problem will give us K bits of Turing's number, and these K bits are not independent pieces of information, because you don't really need to know K yes/no answers, it's sufficient to know how many of the programs halt.

And this is going to be a number between zero and K

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

And if you write this number in binary it's really only about log2 K bits.

$$\#\ \textit{that halt} = \textit{log2 K bits}$$

So knowing how many of programs halt is a lot less than K bits of information, it's really only about log2 K bits, it's the number of bits you need to be able to express a number between zero and K in binary. In this way we get the halting probability, it's defined like this:

Omega Number

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

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

$$0 < \Omega < 1$$

Then write 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.

How can you get this number? 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. If the program p is K bits long, it contributes $1/2^K$; one over two to the K; to this halting probability. 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 : 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. It's incompressible mathematical information. And the reason is, because if you knew the first N bits of this number; 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.

And that means that not only you can't compress it into a smaller

However, you can prove all kinds of nice mathematical theorems about this number, even though it's a specific real number, you can prove all kinds of statistical properties, but you can't determine individual bits!

So this is the strongest version of an incompleteness result . . .

8. Conclusion

Now, we'd like to suggest some questions that we don't know how to answer yet, that are connected, that we think are connected with this stuff that we've been talking about. So let’s mention some questions we don't know how to answer, but we hope that maybe this Chaitin’s stuff is a stepping-stone.

One question is positive results on mathematics:

Positive Results

Where do new mathematical concepts come from?

We mean, Godel's work, Turing's work and Chaitin’s 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 what we need is a more realistic theory that gives us a better idea of why mathematics is doing so splendidly, which it is.

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! 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, and new ideas for building organisms-and the ideas keep getting reused. That's how biology seems to work. Nature is a cobbler! So we think these problems are connected, and we hope they have something to do with the ideas we mentioned here.

9. References

1. G. J. Chaitin, "Conversations with a Mathematician: Math, Art, Science and the Limits of Reason, Sringer-Verlag", London, 2002
2. G. J. Chaitin, "Exploring Randomness", Springer-Verlag, London, 2001
3. G. Polya, "How to Solve It: A New Aspect of the Mathematical Method", Princeton University Press, Princeton, New Jersey, 1988