Gödel's Incompleteness Theorems

Version 4.3

Richard Baron

2 October 2016

Contents

1 Introduction

This paper grew out of a talk for non-specialist audiences. It does not use logical symbols. Instead, it offers ordinary language versions of statements that would be put in symbolic terms for specialist audiences. In section 13 there are recommendations for further reading, in which the presentation is more technical.

1.1 Summary

We start with a note on Kurt Gödel and on the context within which he worked. We then have some scene-setting in which we set out what we mean by symbols, formulae, sentences, numbers and proofs. After that, we can define formal systems. The theorems that concern us are about these systems.

After that, we start to look at Gödel's first incompleteness theorem. This theorem is the main topic of the paper. It tells us that there is an inevitable limit on the power of certain systems of logic to prove things. We start by outlining what the theorem says, and then set out the conditions a system must meet if we are to show that it is limited in what it can prove. After that, we show how to get to a special sentence that will lead us to the theorem and how it does lead to the theorem.

We then turn to the second incompleteness theorem. It tells us that there are powerful and useful systems which cannot prove their own consistency. We set out what the theorem says, and then how to get to it.

Finally, we comment on the significance of the two theorems.

1.2   Availability and licence

This paper is available at the following address:

The author's home page is at the following address:

There is a link to this paper from the "Course Notes" page on that website.

1.3   Comments, changes and acknowledgements

This paper is intended to help people who want to learn about the theorems. Comments are welcome. Please send them to the address given at:

http://www.rbphilo.com/contact.html

Any revised version will have a new version number and a new date. In addition, changes will be noted in the list of changes in section 14 at the end of the paper.

2 Kurt Gödel

2.1 Life

Kurt Gödel was born in 1906 in Brno, then in the Austro-Hungarian Empire but now in the Czech Republic. He went to the University of Vienna to study physics, but he also worked on the foundations of mathematics and on related philosophical issues. His doctorate, on logic, was awarded in 1929.

In 1931, he published the two incompleteness theorems that will concern us. During the 1930s, he made visits to the Institute for Advanced Study at Princeton. He was back in Austria in 1939, but he left after the Second World War started and travelled back to Princeton. He published further important results in logic in the 1940s. He regularly conversed with Einstein, who was also at Princeton, and he discovered a new solution to Einstein's field equations, which give the possible shapes of universes. He remained at Princeton for the rest of his life, and died in 1978.

His work as a logician is hugely important. He would have to be included in any list of the top six logicians of all time. And he would be a very strong candidate to be placed in the top three, alongside Aristotle $(384 - 322$ BC), whose works provided the foundations of logic as a systematic discipline, and Gottlob Frege $(1848 - 1925)$, who was the chief architect of the modern form of logic.

2.2 The mathematical context

In various parts of mathematics we see how we can start with some basic principles, and then establish sophisticated conclusions that follow inevitably from those principles.

For example, we can start with some basic principles of geometry, Euclid's axioms, and prove Pythagoras' theorem.

It is an obvious aspiration to extend this work to cover mathematics as a whole. It would be good to identify some basic principles, principles that it would seem impossible to deny, and then to show that all of mathematics could be built on that firm foundation.

The most ambitious attempt in that direction was published while Gödel was still a child. In 1910, 1912 and 1913, Bertrand Russell (1872 − 1970) and Alfred North Whitehead (1861 − 1947) published the three volumes of *Principia Mathematica*. But while this work accomplished a great deal, and showed that at the very least, large parts of mathematics could be founded on some basic principles, it was clear that not everything that needed to be done had been done.

In 1921, while Gödel was in his teens, David Hilbert (1862 − 1943) set out what he thought still remained to be done. He proposed a programme to set out foundations for the whole of mathematics, such that we would have no fear that the foundations might be in the least bit shaky.[1] On the one hand Hilbert wanted everything covered, and Russell and Whitehead had not done that much. On the other hand, Hilbert had a specific concern about the firmness of whatever foundations were put in place. While the concept of infinity had been in use since ancient times, the rigorous treatment of infinite quantities was quite new, and it was not yet clear whether their use might take mathematics into territory in which there was a risk of error.

---

[1] Hilbert had been concerned about such issues since the start of the century. He was not initially motivated by the work of Russell and Whitehead. But it was in some talks in Hamburg, in July 1921, that he set out the definitive version of his programme.

It was in this exciting context that Gödel set to work. As we shall see in section 12.1, his two incompleteness theorems presented a major challenge to Hilbert's programme. The programme cannot be carried out in full. There must be gaps, and we cannot be completely certain of the firmness of the foundations.

3 Symbols, formulae, sentences and numbers

We are going to be concerned with formal systems – we shall just say **systems** for short. Within a system, we use symbols to write down formulae. Some of the formulae will be sentences. Quite a lot of the formulae will involve numbers. And there will be proofs of sentences. In this section we shall explain what we mean by symbols, formulae, sentences and numbers. In section 4 we shall introduce the idea of a proof. In section 5 we shall explain what counts as a system, and we shall be precise about what counts as a proof.

3.1 Symbols

We write things down using **symbols**. Here are some examples of symbols:

- Mathematical symbols like 5, + and =.

- Ordinary letters when they are used on their own, rather than in words. Examples are x and y in "$y = 2x$".

- Ordinary words.

If we look at the systems that logicians use, we find that they do not count ordinary words as symbols. Logicians replace the words that they might need with special symbols, and they do not use other words. This is partly because they do not want any of the ambiguity of ordinary language to creep into their systems, and partly because they want to be absolutely clear about which symbols can be used in each system. But we shall give examples in ordinary language, so we shall treat ordinary words as symbols.

3.2   Formulae

We put symbols together in strings, by writing one symbol after another along a line. Not just any old string is acceptable. Most random strings of symbols are gibberish. We are really interested in strings that comply with grammatical rules about what may be written.

A string that complies with the rules is officially called a well-formed formula, or a wff. But we shall simply call such a string a **formula**.

Each formula is written on its own line. Here are some examples of formulae:

- Yeats was a poet.

- 31 is a prime number.

- x is a prime number.

Formulae can be very long, but we want to maintain the principle that each formula occupies only one line. If we want to think about writing down a long formula, we simply imagine that we have a very wide sheet of paper. And if any formula in this paper strays onto a second line, the fact that it is really a single-line formula will be shown by the fact

that there will be a small gap below it, like a break between paragraphs.

We often display sequences of formulae, one after another. We do this when we set out a proof. We put each formula on a separate line. We also quite often number the lines, just to make it easy to refer to lines in the proof, but a line number does not become part of the formula on that line.

Here is an example, a proof that the law forbids smoking in the dining room of a restaurant. Once we have been given lines 1 to 4, we can work out that lines 5 to 7 follow from them. Line 5 follows from lines 2 and 3, line 6 follows from lines 4 and 5, and line 7 follows from lines 1 and 6.

1. The law forbids smoking in enclosed spaces that are open to the public.

2. The law defines rooms in buildings as enclosed spaces.

3. A dining room in a restaurant is a room in a building.

4. A dining room in a restaurant is open to the public.

5. A dining room in a restaurant is an enclosed space.

6. A dining room in a restaurant is an enclosed space that is open to the public.

7. The law forbids smoking in the dining room of a restaurant.

3.3 Sentences

3.3.1 Formulae that are sentences

A formula may qualify as a sentence. A **sentence** is a formula that makes a definite statement. It may be written in ordinary language, or it may be written in mathematical symbols. So the following are sentences:

- Masha bakes pies.

- $11 + 6 = 17$

- $2 + 2 = 5$

- Not $(2 + 2 = 5)$

The third one of these, "$2 + 2 = 5$", is a perfectly good sentence. But it is not often useful.

3.3.2 Negations of sentences

The last example above, "Not $(2 + 2 = 5)$", is the opposite of the third one. It denies that $2 + 2 = 5$. Sticking the word "Not" on the front of a sentence is the way in which we say the opposite of what the sentence on its own would say. The resulting sentence is called the **negation** of the sentence with which we started.

We shall always write "Not" with a capital letter when it plays this role, just to remind us of what it is doing. We shall also wrap the original sentence in brackets before we stick "Not" on the front, so as to make it clear exactly what is covered by the "Not". We shall do this even when we give a sentence a one-letter name, like G, so we shall write the negation as "Not(G)". When a sentence is given a one-letter name, we shall eliminate the space after the "Not", just to emphasize the link between the "Not" and the sentence.

The negation is, however, not the extreme opposite. It is just the other option. If we start with "John is tall", and we form the sentence "Not (John is tall)", this does not necessarily mean that John is short. He might be short, or he might be of middling height. We have only denied that he is tall.

3.3.3   Formulae that are not sentences

A formula that does not make a definite statement is not a sentence. For example:

- x is greater than 1,000.

- z is a philosopher.

These do not make definite statements. "x" stands for some number or other, but we are not told which number. We are invited to select a number. The result may be correct (if, for example, we select the number 2,345), or it may be incorrect (if, for example, we select the number 738). "z" stands for some person or other, but we are not told which person. We are invited to select a person. The result will be correct if the person we select is a philosopher, and incorrect if he or she is not a philosopher.

We can form negations of formulae that are not sentences, in the same way that we form negations of sentences. We simply put "Not" on the front of a formula. "Not (z is a philosopher)" again invites us to select a person. The result will be correct if the person we select is not a philosopher, and incorrect if he or she is a philosopher.

3.4 Numbers

We are going to mention numbers quite a lot. We shall always mean the numbers 0, 1, 2, 3, and so on. We shall not mean fractions, negative numbers, infinite numbers or anything else.

4 Proofs of sentences

Some sentences have proofs. The rough idea of a proof may be familiar from school mathematics or from court cases. In school, you may start from some equations and prove that one value is twice another; or you may start from some basic ideas about geometry and prove that the angles of a triangle add up to 180 degrees. In court, a lawyer may start from some sections of a law and prove that some type of conduct is not allowed.

To prove a sentence, we start from some sentences we are already given. They may be equations, or ideas about geometry. Or they may be sentences like the first four sentences in our example in section 3.2 ("The law forbids smoking in enclosed spaces that are open to the public", and so on). These starting points are called premises. Then we work through a sequence of sentences to reach the sentence we want to prove. This sentence is called the conclusion. It might be something mathematical, or it might be a conclusion like "The law forbids smoking in the dining room of a restaurant". All of the steps through the sequence must be strong ones, so as to convince us that the conclusion does indeed follow from the premises.

We are only going to be interested in proving mathematical sentences. This will allow us to be precise about what counts as a proof. We shall be precise in section 5.2.

5 Systems

5.1 Systems are boxes of tools

A **system** is a box of tools for writing and proving sentences. There are four types of tool in the box.

Tools of the first two types are there to help us write sentences. They are as follows:

- There is a collection of symbols we can use, such as letters, numbers, words, the + sign, the = sign and brackets.

- There are tools to help us to combine symbols. There are some rules that tell us which strings of symbols count as formulae. These are the only strings we are allowed to write. There are also rules that tell us which formulae count as sentences, and how we can generate new formulae and sentences from ones we already have. For example, there may be a rule that if we have two sentences, we can always create a new sentence by joining them together with the word "and". That rule would mean that if the rules allowed us to write "The bear wants to sit down", and the rules also allowed us to write "The bear wants to eat one of Masha's pies", we would be allowed to write "The bear wants to sit down and the bear wants to eat one of Masha's pies".

Tools of the third and fourth types are there to help us prove sentences. They are as follows:

- There are some sentences called **axioms**, which we are given for free. We do not need to prove them, so we can always use them as premises in arguments.

- There are some **rules of inference** that tell us how we can move step by step through a proof, by going from formulae we already have in the proof to new ones.

For example, we can use the following axioms for the numbers, 0, 1, 2, 3, and so on:

1. There is a number, which is called 0.

2. Each number has a successor, the number that comes next. (1 is the successor of 0, 2 is the successor of 1, and so on.)

3. 0 is not the successor of any number. (This means that 0 is the first number in the chain of numbers.)

4. If any two numbers have the same successor, they are the same number. (This means that the chain of numbers cannot loop back on itself. We cannot go 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 5, because then 9 and 4 would have the same successor, 5, so they would have to be the same number.)

5. Suppose that 0 has a property, and suppose that if any number has a property, so does its successor. Then all numbers have it. (What this says is that if we can find a property that applies at the start, and that always gets passed up the chain, every number has the property. It is actually a pattern for an axiom. There are infinitely many specific axioms, one for each specific property.)

What about the rules of inference? They tell us what moves we can make in order to work our way forward from formulae we already have to new formulae, along a sequence that will eventually get us to a sentence we would like to prove. We shall only look at one rule of inference, just to give us the idea. This is the most important rule of inference, and we find it in most systems. It is called **modus ponens**:

- If we already have a sentence, P,

- and we already have the sentence, If P, then Q,

- we can move on to the sentence, Q.

For example:

- If we already have "12 is an even number",

- and we already have "If 12 is an even number, then 12×3 is an even number",

- we can move on to "12×3 is an even number".

That is our complete box of tools. We shall speak of writing formulae, and of giving proofs, *within a system.* This will mean using only the tools that the system has, and writing formulae and giving proofs which we shall take to belong to that system. This will be so, even if identical formulae could be written, and identical proofs could be given, in other systems.

There can be different boxes of tools. Boxes may differ in respect of their symbols, their rules about what we can write, their axioms, or their rules of inference. Each box will be a different system. One analogy is different systems for proving things in court, in England and in France. The available symbols and rules for combining those symbols are different because English and French have different vocabularies and different grammars. And the available starting points in argument that lawyers can use without

needing to justify them are different because those starting points are the provisions of the laws of the two countries, and those laws are different.

5.2 Proofs within systems

Once we have our tools, we can prove lots of sentences.

For example, once we have all the axioms for numbers we gave above, along with some other axioms that are not specific to numbers but are standard for the appropriate type of system, we can use the axioms as premises. Then we can give proofs of sentences like these:

- $5 + 7 = 12$

- Not $(3 + 4 = 8)$

Note the second example. It is important that we can prove some negations, that is, some sentences which start with "Not", as well as other sentences which are not negations.

Now we can be precise about what counts as a **proof** of a sentence. It is a sequence of formulae, where each formula is:

1. an axiom; or

2. something that our rules of inference allow us to move on to, given the earlier formulae in the sequence,

and the sequence ends with the sentence we want to prove.

We are going to get very interested in which sentences have proofs, and which ones do not have proofs, in a given system. We shall speak of a system allowing us to prove a sentence. This will mean that there is a proof of the sentence *within the system.*

If there is no proof within a given system, that will not mean that there is no proof anywhere. There might be a proof in another system. One system may, for example, be able to prove more than another one because it has more axioms. But we are going to be concerned with what particular systems can do. This means that we should not speak of a sentence being provable in general. We should only speak of its being provable within some particular system.

6 The first incompleteness theorem

We like to think that if a system allows us to write a mathematical sentence, the same system should also allow us to prove it or to prove its negation (but not both of them). So if we are faced with these two:

- The sum of any two even numbers is an even number;

- Not (The sum of any two even numbers is an even number),

we should be able to prove the first one − and indeed we can, so long as the system has enough resources to generate the proof. We also find that we cannot prove the second one.

Likewise, if we are faced with these two:

- 22 is greater than 45;

- Not (22 is greater than 45),

we should be able to prove the second one − and indeed we can, so long as the system has enough resources to generate the proof. We also find that we cannot prove the first one.

It would be a bit worrying if there were a mathematical sentence, such that a system allowed us to write it but did not allow us to prove it, and did not allow us to prove its negation either. What should we do? We could not accept both the sentence and its negation, because they would clash. We would feel very uncomfortable accepting neither of them, because that would mean not knowing the answer to a mathematical question, and we tend to think that all mathematical questions should have answers. But if we were to accept just one of them, we would have to do so without the reassurance of a proof.

Gödel's first incompleteness theorem tells us that we will routinely be faced with sentences just like that. To be precise, it says the following.

Suppose we have a system that meets the following conditions:

1. The system is consistent.

2. We can identify formulae, sentences, axioms and proofs.

3. The system is powerful.

(We shall explain what these conditions mean in section 7.)

If the conditions are all satisfied, there will be a troublesome sentence, which we shall call G, with the following properties:

1. We can write G within the system.

2. The system does not allow us to prove G.

3. The system does not allow us to prove Not(G).

How do we get to G? We have to find a route to a sentence which will claim that it has no proof. This sentence will be our G, so G will talk about itself. (The lack of a proof

will be the lack of a proof *within the relevant system.* There may be a proof in some other system that what G says is correct.)

As we shall see in section 9, we can get to a sentence like that. And in section 10, we shall see how that sentence gets us to the first incompleteness theorem. If we work through all that, we shall get a very good idea of what the theorem says. Then we shall be able to grasp its significance.

One thing we can explain now is our use of the word "incompleteness". The concept is actually the concept of negation incompleteness. A system is negation incomplete if there is some sentence which the system allows us to write, but which is such that the system does not allow us to prove it, and does not allow us to prove its negation either. (There might be just one sentence like that, or there might be several of them.) A system is negation complete if for every sentence which it allows us to write, it allows us to prove either the sentence or its negation. We say "incomplete" and "complete" as short forms of "negation incomplete" and "negation complete".

(Logicians also use the words "complete" and "incomplete" in another sense. In this other sense, a system is complete if there are proofs of all of the sentences that have to be true. Otherwise, it is incomplete. The context usually makes it clear which sense is meant. We are only going to discuss negation completeness and incompleteness. We shall not use the other sense.)

7   The conditions on the system

In this section, we shall set out the conditions that a system needs to meet in order for us to prove that the system is incomplete. We should note at the outset that they are

not arbitrary conditions, rigged so as to allow us to prove incompleteness. They are conditions that tend to make systems useful. It would not be at all surprising to find that a system in use met these conditions.

7.1 The system is consistent

A system is consistent so long as there is no sentence S, such that we could use the system to give a proof of S *and* a proof of Not(S). If there is a proof of some sentence S, and a proof of Not(S), that means the system contains a contradiction. Then we say that the system is inconsistent.

We shall illustrate the notion of a contradiction by considering an example. We shall then set out why consistency matters.

Suppose that a system allowed us to prove both of the following sentences:

- Every even number apart from 2 is the sum of two prime numbers.

- Not (Every even number apart from 2 is the sum of two prime numbers).

The first one is called Goldbach's conjecture. It looks plausible. 4 is $2 + 2$, 6 is $3 + 3$, 8 is $3 + 5$, 10 is $3 + 7$, and so on. It has been checked up into the billions of billions, but there is no proof yet. The second one is the negation of Goldbach's conjecture.

What would happen if the system that we normally used for numbers allowed us to prove both of these sentences?

We would panic. We would have a contradiction, and that is disastrous for any formal system. Apart from anything else, if a system allows even one contradiction, then we can use

the system to prove any sentence at all which the system allows us to write. We would end up proving that $2+2=5$. So inconsistent systems are not much use.

Inconsistent systems cannot be incomplete. Incompleteness arises when a system allows us to write a sentence, but not to prove either it or its negation. In an inconsistent system, we would be able to prove both the sentence and its negation, because the system could be used to prove anything that it allowed us to write.

7.1.1 Consistency and ω-consistency

There is a little twist here. We are taking the route to the first incompleteness theorem that Gödel originally used, because that makes it easy to understand the special sentence G. If we take that route, consistency is not quite enough. We actually need something a bit stronger, called ω-consistency.[2]

There are two reasons why this is not a major issue for us, given that we are taking a fairly informal approach. The first reason is that the basic idea is the same: ω-consistency, like consistency, amounts to not being able to prove some dubious combinations of sentences. The second reason is that there is another route to the first incompleteness theorem, formulated by John Barkley Rosser $(1907-1989)$. He published his route in 1936. It gets us there while only assuming consistency, and not requiring ω-consistency. The catch is that the sentence Rosser used to do the work that Gödel's sentence G does, is more complicated. It is harder to see how Rosser's sentence does the job.

---

[2] The character ω is the lower-case version of the Greek letter Omega, a long o. In English it is pronounced "omega", not "o".

Having said that, here is a note on ω-consistency.

Suppose that a system is concerned with a collection of objects, for example, all the numbers 0, 1, 2, 3, and so on. And suppose that for each object, we can prove within the system that it lacks some particular property. Then for the system to be ω-consistent, we must not be able to prove within the system that there exists an object which has the property.

The systems that normally interest us are all ω-consistent, so far as we can tell. This makes it hard to give a realistic mathematical example of its absence, that is, of ω-inconsistency, so as to illustrate the kind of dubious combination of provable sentences that we want to avoid. So here is an (unrealistic) real-world example.

Suppose that we have genetically modified a strain of apple trees, so that the modified trees will never produce green apples, but only red ones. Then we plant an infinite number of the modified trees.

By inspecting the genome of any tree, we could prove that it would lack the property of producing green apples.

If we could also prove that some tree or other (we could not say which one) would produce green apples, we would have a logical problem. There would be ω-inconsistency. But so long as we could not prove that, there would be no problem. There would be the nice safe ω-consistency.

Finally, we say that ω-consistency is stronger than consistency because we first check to see whether systems qualify as consistent. Then we look among the consistent systems to pick out the ω-consistent ones. It is only consistent systems that can qualify as ω-consistent. So if a system is ω-consistent, it is also consistent. But it is possible for a system to be consistent without being ω-consistent. Looked at another way, if a system is inconsistent, it must also be

ω-inconsistent. But a system can be ω-inconsistent, while still being consistent.

7.2 We can identify formulae, sentences, axioms and proofs

It must be possible to work out the status of a string of symbols, or of a sequence of strings of symbols. We must always be able to answer the questions that we cover in this section.

7.2.1 Is a string of symbols a formula, and is it a sentence?

There must be a mechanical procedure that will do both of the following, and will do so within a finite number of steps:

- If we apply the procedure to any finite string of symbols that is a formula, it will work out that the string is a formula. If the string is not merely a formula, but a sentence, it will work that out too. And if the string is a formula which is not a sentence, it will work out that the string is not a sentence.

- If we apply the procedure to any finite string of symbols that is not a formula, it will work out that the string is not a formula.

Normally, this is not a problem. But we can imagine that it might be a problem, for example if there were some lack of clarity in the rules that set out what counted as a formula. So we need to record the condition.

A mechanical procedure is one that we specify in advance, in such detail that once we have specified it, it can be applied simply by following the instructions. There must not be

any need for someone applying the procedure to exercise judgement, or even common sense. And the procedure must not be modified as work on a string progresses. (The procedure can include conditional instructions within it, like "If the string of symbols starts with a 'Not', do X, but if there is no 'Not' at the start, do Y instead". But all of those conditional instructions must be written into the procedure before anyone starts to use it.) This definition of a mechanical procedure applies to the two following conditions as well.

This condition, like the two following ones, also refers to reaching a conclusion in a finite number of steps. We need to understand what steps are, and we also need to understand just how generous the limit is.

We can grasp the idea of a step by thinking of a particular method we might use to check whether a given string was a formula. We might look at the rules for building up formulae, and see how they allowed us to build up complicated formulae from simple ones. Then we could work out a way to generate formulae in some order, starting with the simplest ones. Then we could implement this method, and start to generate formulae. Each time we generated a new formula, we could compare it with the string that was in front of us. If there was a match, the string would be a formula. If we could tell when we had gone so far through the formulae that we must have passed the point where the string might have been, then it would not be a formula.

We can see what the steps would be in this sort of procedure. Each act of adding to a formula to generate a new one would be a step. For example, if we already had a formula, creating its negation by putting "Not" on the front of it would be a step. Or if we already had two formulae, combining them by putting "and" in between them to generate a longer formula would be a step. In the

comparison, comparing each formula with the string under consideration would be a step. Deciding "No match, try the next formula" would be a step. And so on.

We only require that the number of steps needed to work out whether a string is a formula, or whether it is a sentence, should be finite. It could be a hundred steps, or it could be trillions of trillions of steps. It might take us longer than the remaining life of the Universe to get through them all, but that would not matter. Logicians do not worry about practical details like that. All that matters is that whatever the string, the number of steps should be finite. (All of this also applies to the mentions of a finite number of steps in the next two conditions. The number of steps to determine whether a sentence is an axiom, or whether a sequence of formulae is a proof, must be finite whatever the sentence, or whatever the sequence of formulae.)

7.2.2   Is a sentence an axiom?

There must be a mechanical procedure that will do both of the following, and will do so within a finite number of steps:

- If we apply the procedure to any sentence that is an axiom, it will work out that the sentence is an axiom.

- If we apply the procedure to any sentence that is not an axiom, it will work out that the sentence is not an axiom.

Again, this is not normally a problem. If we have a finite number of axioms in the box of tools we just go through a list of them, and compare each axiom on the list with the sentence we have in front of us. If it matches one of the axioms on the list, we will notice this and say that it is an axiom. If we get to the end of the list of axioms without finding a match, we will say that it is not an axiom.

But it can be a problem. There are systems with infinite numbers of axioms. In fact, some of the systems that we find interesting, including the standard system to handle the arithmetic of the numbers 0, 1, 2, 3, and so on, do have infinite numbers of axioms. In such a system, if we took a sentence that was in fact an axiom, and started to work through the axioms systematically, we should eventually discover that it was an axiom. But if it was not an axiom, we might go on for ever, always hoping that our sentence would turn up among the axioms but never discovering that it was not an axiom.

We might not get caught in this trap. If for example there was a list that arranged the axioms in some order, and we knew that if our sentence was an axiom, it would have to come before a certain point in the order, we could stop once we had reached that point. Alternatively, if there were a finite number of patterns of axiom, so that any sentence which followed one of those patterns was an axiom but no other sentences were axioms, we might be able to check whether the pattern of the sentence in front of us matched one of the axiom patterns. And in fact, we do not get caught in the trap when we use the standard system for arithmetic.

However, there are systems in which we would be caught and would not have one of these helpful escapes from the trap. So we need to record the condition that we can always work out, in a finite number of steps, whether any given sentence is an axiom.

7.2.3   Is a sequence of formulae a proof?

There must be a mechanical procedure that will do both of the following, and will do so within a finite number of steps:

- If we apply the procedure to any finite sequence of formulae that is a proof, it will work out that the sequence is a proof.

- If we apply the procedure to any finite sequence of formulae that is not a proof, it will work out that the sequence is not a proof.

This should not be a problem. We have already laid down the condition that we can tell whether any given sentence is an axiom, so we can go through any purported proof and pick out all the axioms. Then we can see whether all the other formulae are ones that can be made to follow from the axioms in the proof by using the rules of inference.

So long as we have a finite number of rules of inference, we can start from the beginning of the sequence of formulae and try all the possible ways to derive each non-axiom formula in turn from what precedes it in the sequence of formulae. If we find a way that works, then the formula can be made to follow. If we try all of the ways and find that none of them works, the formula cannot be made to follow and the sequence of formulae is not a proof. If we go through all of the non-axiom formulae in this way, and find that each one can be made to follow from what precedes it, then the sequence of formulae is a proof. We can imagine difficulties if there were an infinite number of rules of inference to try, but that would make for a very odd system.

Given that there should be no difficulty in working out whether a sequence of formulae is a proof, it might seem odd to bother specifying this condition. But there is a reason for doing so. A system will itself need to be able to say whether

there is any sequence of formulae that is a proof of a given sentence. So we need to be confident that picking out proofs is a manageable task, one that could be performed within the system. For the same reason, we need to be confident that it is a manageable task to pick out the things that go into proofs, that is, formulae, sentences and axioms.

7.3   The system is powerful

A system is powerful if it can handle the arithmetic of the numbers 0, 1, 2, 3 and so on. There are systems without that level of power, and they do allow us to do some very useful mathematics. Some of those systems do not suffer from incompleteness. But we shall concentrate on systems that do allow us to count in whole numbers, say when computations have whole-number answers, and useful things like that.

We shall first make the notion of a powerful system precise, and then explain why we are going to rely on power when we establish incompleteness.

7.3.1   The notion of a powerful system

A system is powerful if:

- we can apply it to the arithmetic of the numbers 0, 1, 2, 3, and so on, with addition and multiplication; and

- when we do so, it can prove the answers to a wide range of questions. It must be able to prove that the answer to "What is the result of 7×9?" is "63", that the answer to "What is the result of working out whether 317 is a prime number?" is "Yes, it is prime", and so on.

"A wide range of questions" is a vague phrase. The official definition is as follows.

We can ask questions that demand the answers to computations. (The question "What is the result of 7×9?" obviously demands the answer to a computation. This is not so obvious with the question "What is the result of working out whether 317 is a prime number?", but actually, it does demand the answer to a computation. It asks us to calculate whether the answer is "Yes, it is prime" or "No, it is not prime".)

We are allowed to build up the computations that give us our questions in the following ways:

- We can combine three basic operations: going from a number to zero, adding one to a number, and picking a number out from a list.

- We can combine these operations by applying one of them, and then applying another one (or the same one again) to the result, as many times as we like.

- We can also combine them by having a rule that tells us the result of a computation on 0, and then tells us how to work out the result for a computation on any greater number by looking at the result for the previous number. So if we want to know the result for a computation on 3, we look at the result for 2. To discover that, we look at the result for 1. To discover that, we look at the result for 0. We are given that, so we can work back up the chain and find the result for 1, then the result for 2, then the result for 3.

This may not sound like much. But these three basic operations, and two ways to combine them, actually make it possible to reach a remarkably large amount of mathematics. And they allow us to define powerful systems, as follows:

- Take all of the computations that can be built up using these operations and ways of combining them.

- Take all of the questions that demand answers to those computations.

- A system is powerful if, and only if, it can prove the answers to all of those questions.

7.3.2 Why power matters

It might seem odd that we can only establish incompleteness for powerful systems. We might expect weak systems, rather than powerful ones, to find it difficult to prove things. We shall now explain why it is not odd. We shall first consider what systems allow us to write, and then what they have the power to prove.

Incompleteness arises when a system allows us to write a sentence, but does not allow us to prove it or to prove its negation. If we want to establish incompleteness, we first have to show that a system has the resources to write a suitably troublesome sentence.

We shall start by noting one way in which systems can give us great scope to write things. They can do so by allowing us to make generalizations that range across all numbers. Here are examples of generalizations of this sort:

- All prime numbers greater than 2 are odd.

- If any two numbers have the same successor, they are the same number. (This was the fourth one of our axioms for handling numbers, as listed in section 5.1. Note that it is a generalization across all numbers, because it is about *any* two numbers.)

The ability to make generalizations is immensely useful. Imagine how much more laborious life would be if we could not write the first of these examples, and had to write "3 is odd, 5 is odd, 7 is odd, 11 is odd, ... ".

This ability is also going to matter when we come to write our troublesome sentence G. We are going to give identification numbers to proofs, and G will say that no number is the identification number of a proof of itself. So G will make a general claim about all numbers.

Lastly, the ability to make generalizations can allow a system to prove those answers to questions that it needs to be able to prove, in order for it to qualify as powerful.

The ability to make generalizations can lead to incompleteness. But we defined powerful systems as systems with lots of power to prove things. Why does a system need lots of power, in order for us to show that it has limited power to prove things?

To see why we rely on power to demonstrate incompleteness, we must first make a connection with the mechanical procedures that we mentioned in section 7.2. It turns out that if mechanical procedures are available, then computations of answers to questions which ask "Is this a formula?", "Is this a sentence?", "Is this an axiom?" or "Is this a proof?" are all computations of the types that we specified when defining powerful systems. We can build up the computations that are needed to answer those questions by using the three basic operations and two ways to combine them that we specified in section 7.3.1. Computations which have been built up like that can work every time, whether the answers are "Yes" or "No". This means that if a system is powerful, it can prove the answers that would come from all applications of such mechanical procedures.

Now recall that a proof is a proof of whatever sentence comes at the end of the sequence of formulae. This means that if a sequence of formulae is a proof, it is obvious what the sequence proves. So if a system can prove the answer to the question "Is sequence U a proof?", it can also prove the answer to the question "Is sequence U a proof of sentence V?". It only needs to check whether the sequence ends with sentence V.

We are going to consider what would happen if there were a proof of G, or a proof of Not(G). We shall show that in either case, we would get a contradiction. But a contradiction only arises when a system can itself prove both a sentence and its negation. We shall assume that there is a proof of G. Then we shall invoke the power of the system to prove the answer to each question of the form "Is sequence U a proof of sentence V?", to show that on our assumption, the system could itself prove that there was a proof of G. But the claim that there is a proof of G will turn out to contradict what G itself says. We shall then run through a similar argument for the assumption that there is a proof of Not(G), and we shall again invoke the power of the system to make sure that we get our contradiction within the system. We shall derive our contradictions in sections 10.2.1 and 10.2.2.

8 Sentences that talk about sentences

8.1 The problem

When we first mentioned our sentence G, we noted that it will talk about itself by claiming that it has no proof. (Remember that the lack of a proof will be the lack of a proof within the relevant system. There may be a proof in some other system that what G says is correct.) So we need sentences to be able to talk about sentences. More generally, we are going to need formulae to talk about formulae. (Remember that sentences are formulae of a certain type.)

We can manage this in ordinary language. We can write things like:

- The sentence "Though this be madness, yet there is method in't" was written by Shakespeare.

But many systems lack the symbols that we would need to do this sort of thing. In particular they lack ways to name sentences directly, or to put quotation marks around a sentence in order to show that we are talking about the sentence itself. More generally, they are not set up to talk about formulae, sentences, axioms or proofs. These are terms that we use when we talk about systems from outside them, rather than terms that are found within systems. What can we do about this?

The answer lies in the nature of systems of the type that interest us. These systems may not be any good at talking about formulae, sentences, axioms or proofs, but they can be used to talk about numbers, about the properties of numbers, and about relations between numbers. So we do two things. First we label all strings of symbols, and all

sequences of strings of symbols, with numbers. We cover this in section 8.2. Then we move to a parallel world of properties of numbers and relations between numbers. This parallel world can be represented within any system that we might want to show was incomplete. We cover this in section 8.3.

Section 8.2 is essential to follow what comes next. Section 8.3 on the other hand is an optional extra, given our informal approach. After explaining the parallel world we shall revert to talking about formulae, sentences, axioms and proofs, simply because that makes it easier to see how the proof of incompleteness works. The fact that the parallel world really is parallel makes this informal approach acceptable. But the parallel world is absolutely vital for the sort of formal treatment of incompleteness that is found in books that take a mathematical approach.

8.2 Labelling with numbers

We start by giving every string of symbols a number. The string may be an axiom, a sentence that is not an axiom, a formula that is not even a sentence, or a string that is not even a formula.

This number is a label for the string of symbols, just as a passport number is a label for a person. Government officials can use a passport number to refer to a person and to say things about them: "The person with passport number 44019852 lives at 23 Railway Cuttings, East Cheam". Likewise, if a string of symbols has a number, formulae will be able to talk about it. The number for each string of symbols is called its Gödel number.

So suppose that the Gödel number for "Though this be madness, yet there is method in't" was 711,456. Then we could write:

- The sentence with Gödel number 711,456 was written by Shakespeare.

Then we do some more numbering. Every sequence of strings of symbols gets a Gödel number. This means that in particular, every sequence of formulae will have a Gödel number. So every proof will have a Gödel number. This will allow us to use numbers to say whether there are proofs of sentences and to identify those proofs.

For example, suppose that the Gödel number of the sentence "$2 + 2 = 4$" was 8,982, and suppose that there was a proof of this sentence which had Gödel number 172,046. Then we could write both of the following sentences, and they would both be correct:

- There is a number that is the Gödel number of a proof of the sentence with Gödel number 8,982.

- 172,046 is the Gödel number of a proof of the sentence with Gödel number 8,982.

(The first one of these two is a roundabout way of saying that there is a proof of the sentence "$2 + 2 = 4$". If there is a proof, it will have a Gödel number, so there will be a number that is the Gödel number of a proof. If there is no proof, there will be no number that is the Gödel number of a proof.)

The Gödel numbers that we have given so far are ones that we have made up. In practice, we would need a system that would let us work out the Gödel number for any string of symbols or any sequence of strings of symbols. Each string, and each sequence of strings, would need to get a different number. Our system would also have to let us take

any number and work back to the corresponding string or sequence of strings, or work out that it was not the number of any string or sequence of strings. There are plenty of systems to do that, and it does not matter which one is used. We shall not need to know actual Gödel numbers for any strings or sequences of strings. We only need to know that it is safe to assume that Gödel numbers would be available. We shall therefore continue to make up numbers.

8.3 The parallel world

Gödel's act of using numbers to label things opened the way for him to do a lot more. We shall now pause to see how the act of labelling allowed Gödel to move the discussion of any given system to a parallel world inside the system, before we revert to talking about systems from the outside.

If we select a system and then ask questions like "Is this a formula?" and "Is this a proof of that?" in relation to that system, we stand outside the system and look at material that has been written by someone working inside the system. We are in the world of everyday speech about the system. With labelling by Gödel numbers in place, we can move to a parallel world inside the system. In that world there are parallel questions, and ways to answer them.

Suppose that we are given a string of symbols, and we want to know whether it is a formula. Think of that as a question about possession of a property. We ask "Does this string have the property of being a formula?".

In the parallel world inside the system, we do not see a string of symbols. Instead we see a number, the Gödel number of the string. Suppose the number is 7,560. Then the parallel question is "Does 7,560 have the formula-number property?". The number will have the formula-

number property if, and only if, the string of symbols with Gödel number 7,560 is a formula.

We can make this move, and find ourselves in the parallel world inside the system, because it is possible to define the formula-number property in purely mathematical terms, just like the odd-number property that is possessed by 1, 3, 5, 7, 9 and so on, but that is not possessed by 2, 4, 6, 8, 10 and so on.

We could do the same thing for other questions. Instead of asking "Is this a sentence?", we could ask "Does 7,560 have the sentence-number property?". And instead of asking "Is this an axiom?", we could ask "Does 7,560 have the axiom-number property?".

Now suppose we are given a sequence of formulae, U, and the last formula in the sequence is a sentence, V. We want to know whether sequence U is a proof of sentence V. Think of that as a question about a relation between U and V. We ask "Does sequence U bear the proof relation to sentence V?". That is a slightly odd way to put the question, but it will make it easier to see how the question in the parallel world really is a parallel question. If the wording seems outlandish, consider that we could easily ask whether Plato taught Aristotle by asking the question "Does Plato bear the teacher relation to Aristotle?". This example brings out a useful point. There is often nothing in the nature of a relation to make it exclusive. The fact that Plato bears the teacher relation to Aristotle does not rule out Aristotle's having had other teachers as well. Likewise, even if U is a proof of V, there might also be other proofs of V.

In the parallel world inside the system we do not see a sequence of formulae, or a sentence that is the last member of the sequence. Instead we see two numbers. The first one is the Gödel number of the sequence U, and the second one is the Gödel number of the sentence V. Suppose that the

Gödel number of U is 342,148 and the Gödel number of V is 7,438. Then the parallel question is "Does 342,148 bear the proof-number relation to 7,438?".

We can make this move, and find ourselves in the parallel world inside the system, because it is possible to define the proof-number relation in purely mathematical terms, just like the multiple-of relation that 6 bears to 3 ($6 = 2 \times 3$), or that 85 bears to 17 ($85 = 5 \times 17$), but that 20 does not bear to 7 (2×7 is less than 20, and 3×7 is greater than 20). Our parallel question is a question about a mathematical relation between two numbers, just like the question "Does 342,148 bear the multiple-of relation to 7,438?".

A lot of work is needed to show that the move to the parallel world is legitimate. All of following need to be shown:

- We can give mathematical definitions of the formula-number property, the sentence-number property, the axiom-number property, the proof-number relation and any other bits and pieces that are needed.

- Our definitions do give us proper parallels to facts about what may be written in systems. A number must have the formula-number property if, and only if, it is the Gödel number of a formula. A number must have the sentence-number property if, and only if, it is the Gödel number of a sentence. A number must have the axiom-number property if, and only if, it is the Gödel number of an axiom. And one number must bear the proof-number relation to another number if, and only if, the first number is the Gödel number of a sequence of formulae that ends with a sentence, the second number is the Gödel number of that final sentence, and the sequence of formulae is a proof of the sentence. The same requirement to give us proper parallels applies to anything else that is needed.

- Any system that we might want to show suffers from incompleteness is powerful enough to prove the answers to questions about properties and relations of numbers like the questions we have given as examples of parallel questions, and to do so entirely within itself. We discussed the question of power in section 7.3. So long as a system is powerful as defined in section 7.3.1, it will be powerful enough. And now that we have taken a glimpse into the parallel world, we can see that when we said, in section 7.3.2, that a powerful system could prove the answers to questions like "Is this a formula?", "Is this a sentence?", "Is this an axiom?" and "Is sequence U a proof of sentence V?", what we really meant was that it could prove the answers to the parallel questions about the properties and relations of numbers.

Fortunately, all of these things have been shown. So incompleteness can be demonstrated in a completely formal and logically watertight way. And that formal proof takes place within the parallel world.

Since the move into the parallel world is a move from using non-mathematical language to making calculations, and since the calculations are all within the scope of the arithmetic of the numbers 0, 1, 2, 3 and so on, the move is called **arithmetization**. Sometimes the longer term "the arithmetization of syntax" is used to emphasize that it is formal systems that interest us here, rather than anything in the world that they might characterize.

Having set out why labelling allows work to take place within any given system, in the parallel world of numbers, their properties and their relations, we shall now revert to talking about the formulae, sentences, axioms and proofs of a system from outside the system. That will make it easier to see how we get to incompleteness. If we just want to

see how the proof of incompleteness works, it is perfectly acceptable to talk about a system from the outside. And we do not miss anything about the structure of the proof, because the parallel world really is parallel. An informal description of the proof from outside any given system, and the actual proof in the parallel world within the system, move in step. We shall however continue to refer to Gödel numbers. We need to do that in order to see how some crucial moves in the proof work, and so as to make sure that our informal description does keep in step with the actual proof in the parallel world.

9 Getting to G

Now that we have got Gödel numbers in place, we can use whichever system we are investigating to formulate its G.

9.1 Diagonalization

First, we need to learn a funny procedure that is called **diagonalization**. This takes almost-sentences, and turns them into sentences.

An almost-sentence is a formula that would be a sentence, except that it has a place which is marked with an x where a number should go. (It may have several places, each marked with an x, and then the same number must go into each place.) It is an almost-sentence because until we put a number in the place, it does not say anything definite. For example:

- "x is an even number" is an almost-sentence;

- "20 is an even number" is a sentence.

Although an almost-sentence does not say anything definite, it is a formula, so it will have a Gödel number.

If we want to diagonalize, we take an almost-sentence and work out its Gödel number. Then we form a sentence by putting that number into the place marked x (or into all of the places marked x, if there are several of them). The result is the **diagonalization** of the almost-sentence. It will be a sentence, because the place (or places) marked x will have been filled in with an actual number.

So suppose that the Gödel number of "x is an even number" is 247,398.

Then the diagonalization is "247,398 is an even number".

9.2   No number of a proof

Now we can write this almost-sentence, which we shall call F:

- Not (There is a number that is the Gödel number of a proof of the diagonalization of the almost-sentence with Gödel number x).

(The almost-sentence will be written within a system. "A proof" will mean a proof within that system.)

This almost-sentence will have its own Gödel number. Suppose that it is 123,456. (Yet again, we are just making up a number. It does not matter what the number is.)

Plug that number in to create this sentence, which will be our sentence G:

- Not (There is a number that is the Gödel number of a proof of the diagonalization of the almost-sentence with Gödel number 123,456).

9.3 We have got to G

Now look closely at the sentence G that we have just given, and at how we got to it. We can put together the following argument.

1. We got to G by plugging F's Gödel number into F. This means that G is the diagonalization of F.

2. G claims that there is no number that is the Gödel number of a proof of the diagonalization of a certain almost-sentence.

3. If there were a proof, it would have a Gödel number.

4. So claiming that there is no Gödel number, amounts to claiming that there is no proof.

5. The almost-sentence in question is the one with Gödel number 123,456.

6. That almost-sentence is F.

7. So G claims that there is no proof of the diagonalization of F.

8. But G is the diagonalization of F.

9. So G claims that there is no proof of itself.

The last line of this argument is what we really need. A sentence that tells us it has no proof of itself will let us get to incompleteness.

10 Getting from G to incompleteness

Now we have seen how to get to our G, we can move on from it to show that if the system meets the conditions we have laid down:

- there is no proof of G within the system; and

- there is no proof of Not(G) within the system.

Then we shall have established incompleteness, for systems that meet the conditions. The theorem says that in any system in which the conditions are met, there is at least one sentence with no proof of itself or of its negation. So if we can give an example of a sentence with no proof of itself or of its negation, we shall have established the theorem.

10.1 What we still have to do

We might think that having G would be enough. After all, it tells us that it does not have a proof. But in fact, there is more work to do. It is all very well for G to tell us that it has no proof, but what if it is mistaken? It might have a proof, even though it claims not to have one. And what about Not(G)? We have not yet done anything to show that it has no proof.

So we still need to show that if the system is consistent, and the system also meets the other conditions, G has no proof and Not(G) has no proof. We shall now do this, in three stages. We shall first show two routes to inconsistency. Then we shall bring everything together, so as to finish our proof of the first incompleteness theorem.

10.2 Proving the first incompleteness theorem

10.2.1 If G has a proof, the system is inconsistent

Let us go back to our sentence G. Remember that it is the diagonalization of the almost-sentence we called F, and that F's Gödel number is 123,456.

G is:

- Not (There is a number that is the Gödel number of a proof of the diagonalization of the almost-sentence with Gödel number 123,456).

Now suppose that there is a proof of G. It will have a Gödel number.

So there will be a number that is the Gödel number of a proof of G. And the system will itself be able to prove that this number is the Gödel number of a proof of G, because the system is powerful enough to prove answers to questions of the form "Is sequence U a proof of sentence V?". We noted this point about power in section 7.3.2.

The system will therefore be able to prove that there exists a number that is the Gödel number of a proof of G. (Going from a particular Gödel number to the general claim that some such number exists is a very easy step, which any powerful system can take. If we have found a particular unicorn, we have automatically shown that at least one unicorn exists.)

But G is the diagonalization of the almost-sentence with Gödel number 123,456.

So the system will be able to prove this sentence, which we may call H:

- There is a number that is the Gödel number of a proof of the diagonalization of the almost-sentence with Gödel number 123,456.

Now we have what we want. G and H contradict each other, because G is Not(H). But we assumed that we could prove G. And we have just proved H. So on our assumption, the system can prove both a sentence, H, and its negation, G. So if there is a proof of G, the system is inconsistent.

10.2.2   If Not(G) has a proof, the system is inconsistent

We can now see what happens if we suppose that Not(G) has a proof.

We first remark that if a sentence starts with a "Not", and we put another "Not" on the front of the sentence, the two "Not"s cancel each other out. "Not (It is daytime)" tells us that it is nighttime. "Not (Not (It is daytime))" reverses the claim that it is nighttime. So it is daytime after all.

G starts with a "Not", so when we write Not(G), we write something with two "Not"s on the front. We get this:

- Not (Not (There is a number that is the Gödel number of a proof of the diagonalization of the almost-sentence with Gödel number 123,456)).

The two "Not"s cancel each other out. So Not(G) is simply this sentence, the sentence H that we have just met:

- There is a number that is the Gödel number of a proof of the diagonalization of the almost-sentence with Gödel number 123,456.

Now suppose that there is a proof of Not(G), that is, a proof of H.

Once we have stripped off the two "Not"s at the start of Not(G), to create H, we can see clearly what it says. It says that there is a number that is the Gödel number of a proof of the diagonalization of our old friend F, from section 9.2.

But the diagonalization of F is G. So there will be a number that is the Gödel number of a proof of G. We have no idea what this number might be, but there will be one. And as before, the power of the system means that if there is any such number, the system will itself be able to prove that some such number exists.

On the other hand, if there is a proof of Not(G), and the system is consistent, there cannot also be a proof of G. So each number must fail to be the Gödel number of a proof of G. And again, the power of the system means that it will be able to prove, for each number in turn, that it is not the Gödel number of a proof of G.

Putting these last two paragraphs together, we see that some number is the Gödel number of a proof of G, while at the same time, each number must fail to be the Gödel number of a proof of G. And the power of the system means that it will be able to arrive at these two contradictory results within itself.

This means that the system would contain a contradiction. So if there is a proof of Not(G), the system is inconsistent.[3]

10.2.3 The proof

First, take it that whatever system we are considering meets the conditions of allowing the identification of formulae, sentences, axioms and proofs, and of being powerful. If it fails these conditions, there is nothing to prove, because the theorem only tells us that *if* all of the conditions are met, *then* the system is incomplete. The theorem says nothing about systems that fail any of the conditions.

What we shall not assume yet, is that the system meets the condition of consistency. Maybe it does, but we want to explore the relationship between that condition and incompleteness.

Assume that G has a proof in the system. On this assumption, the system would be inconsistent. We showed how the inconsistency would arise in section 10.2.1.

Now assume that Not(G) has a proof in the system. On this assumption, the system would be inconsistent.[4] We showed how the inconsistency would arise in section 10.2.2.

---

[3] To be precise, we only show that the system would be ω-inconsistent, as discussed in section 7.1.1. We do not quite show that the system would be inconsistent. (Remember that a system can be ω-inconsistent while still being consistent, although a system can also be both ω-inconsistent and inconsistent.)

The way we show that the system would be ω-inconsistent is as follows. If we assume that there is a proof of Not(G), there must be some number or other that has the property of being the Gödel number of a proof of G. But each number individually must lack the property of being the Gödel number of a proof of G.

[4] Again, to be precise, it would be ω-inconsistent.

So if there is a proof of G, or a proof of Not(G), the system cannot be consistent, and the system fails the conditions. But there must be a proof of one or other of G and Not(G), in order to give completeness.

Now look at what we have just shown the other way round. Completeness implies inconsistency. So consistency requires incompleteness, so long as the system meets the other conditions. We have proved the theorem.[5]

11 The second incompleteness theorem

11.1 The advantages of consistency

We saw in section 7.1 that consistency is a very good property for a system to have. If a system is inconsistent, it will allow us to prove any sentence that it allows us to write, including "$2 + 2 = 5$" and all sorts of other nonsense. So an inconsistent system is useless.

Recall that a system is consistent so long as there is no sentence S, such that the system can generate a proof of S *and* a proof of Not(S). If there is just one sentence like that, we have inconsistency.

---

[5] Again, to be precise, we have only shown that ω-consistency requires incompleteness. But as mentioned in section 7.1.1, there is another route to the theorem which shows that consistency itself, along with satisfaction of the other conditions, implies incompleteness.

Just to tie up one loose end, if a system is ω-consistent, it is also consistent. So if we take the route we have taken here, and assume ω-consistency to show that there cannot be a proof of Not(G), that same assumption implies that the system is also consistent. We can use that implied assumption of consistency to show that there is no proof of G either.

Since consistency is such a good property for a system to have, it would be reassuring if we could prove that a system we wanted to use was consistent. We might think we could do this by identifying all the sentences it could prove and checking that there was no sentence, such that the system could prove both the sentence and its negation. But there are plenty of systems that can prove infinite numbers of different sentences. If we tried to check all the sentences that such a system could prove, we would never finish the job. We must look for a different style of proof, one that is based on the general properties of a system. But we should not expect to get everything we might want. The second incompleteness theorem tells us that many systems are not able to prove their own consistency.

11.2   What the second theorem says

The second theorem says the following.

Take a system that meets all of the conditions we noted for the first theorem, and explained in section 7, as follows:

- The system is consistent.

- We can identify formulae, sentences, axioms and proofs.

- The system is powerful. (In order to establish the second theorem, the system actually needs to be a little bit more powerful than is necessary to establish the first theorem. We cover this point in section 11.3.)

Then it is not possible to give, within the system, a proof that the system is consistent.

11.3 Getting to the second theorem

Note that just like the first theorem, the second theorem as a whole applies to all systems. If a system fails any of the conditions, the second theorem says nothing about that system's ability to prove its own consistency. If, on the other hand, a system meets the conditions, it follows that it is not possible to formulate, within the system, a proof of its own consistency.

So we do not need to show anything for systems that fail one or more of the conditions. We only need to show that it is not possible to formulate a proof of consistency, on the assumption that a system meets the conditions.

Just as with our treatment of the first theorem in section 10.2.3, we shall assume that whatever system we are considering meets the conditions of allowing the identification of formulae, sentences, axioms and proofs, and of being powerful. But this time, we shall also assume that the system meets the condition of consistency, because that is what we wonder whether the system could be used to prove.

The first step is to establish that systems can express their own consistency. It would be rather boring to show that the reason we could not formulate a proof of a system's consistency within the system was that there was no sentence within the system that could express the system's consistency.

Fortunately, this is not a problem. Pick any one sentence that the system can prove. An axiom would do, because each axiom has a one-line proof, with the axiom being written on the line, but any other provable sentence would do just as well. Call the chosen sentence P, and form its negation, Not(P). An inconsistent system could also prove Not(P), because an inconsistent system can be used to prove any sentence that it allows us to write. A consistent system could not prove Not(P), because then it would be able to prove both P and Not(P), and that would be a contradiction. Suppose that Not(P) has Gödel number 9,876. Then we can express consistency with the following sentence, which we shall call C:

- Not (There is a number that is the Gödel number of a proof of the sentence with Gödel number 9,876).

The next step is to go back to our sentence G for the relevant system, which is as follows:

- Not (There is a number that is the Gödel number of a proof of the diagonalization of the almost-sentence with Gödel number 123,456).

We recall that this sentence, G, asserts that there is no Gödel number of a proof of G itself. And we saw in section 10.2.1 that if the system is consistent, there is indeed no Gödel number of a proof of G.

But the claim that there is no Gödel number of a proof of G is what G itself says.

So consistency implies G.

Now suppose that the system could be used to prove its own consistency. Then since consistency implies G, a proof of consistency given within the system could be used to prove G within the system.[6]

But the first theorem has told us that if the system really is consistent, it is impossible to use the system to prove G.

So if the system is consistent, it cannot be used to prove its own consistency.

We have arrived at the second incompleteness theorem.

We may add that even if a system could be used to prove its own consistency, that would not in itself give us much reassurance, because an inconsistent system could prove anything, including the false claim that it was consistent. Having said that, this does not make the second theorem uninteresting. The second theorem establishes a limit on the power of consistent systems which meet the other conditions. And if we were worried that a system might be inconsistent, but we had not found any contradictions, or positive evidence that contradictions existed, we might reasonably suppose that the system was consistent. In that case, the second theorem would tell us not to worry that we could not use the system to prove its own consistency.

---

[6] To be precise, a proof of consistency within the system could be used to prove G within the system, so long as we could also show two things *within the system*, rather than by talking about the system from outside it.

The first thing to show would be that our sentence C, which represents consistency, implied our sentence G. The second thing to show would be that there would be inconsistency if it were possible to prove G within the system. Fortunately these two things can be shown within systems, but demonstrating this would require getting quite mathematical.

We should also note that the requirement to be able to show these things within the system means that systems do need a bit more power in order for us to show that they cannot prove their own consistency, than they need in order for us to show that they are incomplete.

11.4 Could one system prove another's consistency?

The second theorem shows that a system which meets the conditions cannot prove its own consistency. But we do not have to trap ourselves within a single system. Indeed, we have spent much of this paper talking about systems from the outside. So could we stand outside a system, and use some other system to prove the first system's consistency? The prospects are not bright.

11.4.1 Weaker and stronger systems

We shall find it useful to have the notions of weaker and stronger systems. One system is stronger than another if the stronger one can prove all of the sentences that the weaker one can prove, and some extra sentences. If one system is stronger than another, the second system is weaker than the first.

One system may be stronger than another because it has all of the axioms of the weaker system, plus some extra axioms. More axioms mean more ways to start proofs.

Note that the stronger system must be able to prove *all* of the sentences that the weaker system can prove. If one system can prove 99 per cent of the sentences that a second system can prove, plus twice as many other sentences again, the missing 1 per cent means that the first system is not stronger than the second one, and the second one is not weaker than the first one. Instead, we say the the first system is weaker than the second system in some ways, and stronger in others. It is weaker in the ways that give rise to the missing 1 per cent, and stronger in the ways that give rise to the extra twice as many sentences again.

Suppose that we are interested in proving the consistency of a particular system, which we shall call Oursystem.

We might try to use a weaker system, which we shall call Weaksystem.

We might try to use a stronger system, which we shall call Strongsystem.

Finally, we might try to use a system that was weaker than Oursystem in some respects, and stronger than Oursystem in other respects.

11.4.2   A weaker system

Suppose that Weaksystem could prove the consistency of Oursystem. Then Oursystem would be able to replicate the proof within itself, because it could do everything that Weaksystem could do, and more. So Oursystem could itself prove its own consistency. And as we have just seen, that would not be possible unless the system was actually inconsistent (or failed one of the other conditions, but we assume that we would have checked that it passed those conditions).

11.4.3   A stronger system

Now suppose that Strongsystem could prove the consistency of Oursystem. This would be perfectly possible. After all, Strongsystem can do more things than Oursystem can do.

But then, what about the consistency of Strongsystem? As we strengthen a system we give it power to prove more sentences, without its losing its power to prove any of the sentences that it could prove before being strengthened. So our fears about possible inconsistency would not reduce,

and they might increase. The more that a system can prove, the greater should be our fear that there would be a sentence S, such that the system could prove both S and Not(S). So if we could not be confident that Oursystem was consistent, we certainly could not be confident that Strongsystem was consistent. And if Strongsystem were inconsistent, any proof that it gave of the consistency of Oursystem would be worthless, because an inconsistent system can prove anything that it allows us to write.

Moreover, if Oursystem were consistent, we could not use Oursystem to prove the consistency of Strongsystem. If Strongsystem were consistent, that would mean that there was no sentence S, such that Strongsystem could prove both S and Not(S). But then, Oursystem could not have such a dangerous S either, because it could prove less than Strongsystem. So if Oursystem could prove the consistency of Strongsystem, it could prove its own consistency, and we already know that this would not be possible unless Oursystem were inconsistent.

11.4.4 A system that is neither weaker nor stronger

We might try to prove the consistency of Oursystem by using a system that was neither straightforwardly weaker than Oursystem, nor straightforwardly stronger. It could be weaker in some respects, and stronger in others. Indeed, there are approaches like this. But while they give some reassurance by narrowing the range of possible worries, they do not give proofs that put the consistency of systems beyond doubt.

11.5 Gaining reassurance from general remarks

One last option would be to do without a proof of consistency and make do with some general remarks, outside any formal system, which would at least be reassuring.

We can do something here. There is no specific reason to fear that standard systems are inconsistent, only a general worry that they might be. Moreover, people have established a great many elaborate mathematical results using standard systems, and they have not so far uncovered contradictions. There are also arguments which rely on the fact that simple proofs look perfectly safe, and that complicated proofs depend on simpler ones in ways that should not lead us to worry about things starting to go wrong as we progress from the simple to the complicated.

These considerations indicate that we should not be overly concerned at the implications of the second incompleteness theorem. But we should note what it says. Like the first theorem, it reminds us that some powerful and useful systems of logic cannot do everything that we might hope they could do.

12 The significance of the two theorems

The significance of Gödel's incompleteness theorems is still debated, over 80 years after their first publication. We can however say something about each of three topics: projects to use a single set of axioms to provide an undoubtedly secure foundation for all those parts of mathematics that can be put in terms of the behaviour of the numbers 0, 1, 2, 3, and so on; the relationship between provability and truth; and misinterpretations of the theorems.

12.1 Secure foundations for everything

In section 2.2, we outlined Hilbert's programme. He wanted to find a set of axioms that would provide foundations for the whole of mathematics. He also wanted us to have complete confidence that the foundations were firm. Now we can consider the impact of Gödel's incompleteness theorems on Hilbert's programme.

There is some dispute about the precise contents and implications of Hilbert's programme, and therefore about the precise impact of Gödel's work on that programme. But Hilbert certainly wanted a proof of the consistency of a system, where that system would suffice for the whole of mathematics. In addition, some authors argue that his programme would require an absence of negation incompleteness. Regardless of exactly what would fulfil Hilbert's programme, these are both understandable ambitions. Where does Gödel's work leave them?

In order to understand the impact of Gödel's incompleteness theorems, we shall concentrate on the mathematics

that is based on the behaviour of the numbers 0, 1, 2, 3, and so on.

This part of mathematics goes far beyond everyday sums. A huge body of sophisticated mathematics, including important results about prime numbers and about which equations have solutions in the numbers 0, 1, 2, 3, and so on, has been built upon the same foundation as the one that underpins our everyday sums. We shall call this body of mathematics "arithmetic", its traditional name in this sort of discussion. (There is no implication that we are limited to the arithmetic that everyone learns at school; and there is an alternative name, "number theory".)

If we were only concerned with everyday sums, we would have no worries at all. We know that the system works, that every sum has precisely one right answer, and that no contradictions arise. If something goes wrong, it is the fault of the person or the computer making the calculations, not of the mathematics. But can we feel so satisfied, and so secure, about the much more sophisticated results of arithmetic in the wider sense that we have just identified?

In order to see the limits that Gödel's theorems impose, we must ask two questions. First, can we find foundations for the whole of arithmetic? And second, can we have complete confidence in the foundations?

12.1.1 Foundations for the whole of arithmetic

One message of the first incompleteness theorem is that we cannot get every question answered by any one system of axioms. There will always be some incompleteness, so long as the system meets the conditions that we laid down in section 7, the conditions of consistency, identifiability of formulae, sentences, axioms and proofs, and power. And we would really like any system to meet those conditions.

It is no surprise that a given system may be inadequate to our needs. Indeed, mathematicians have identified results that can be expressed in the system that we ordinarily use for the arithmetic of the numbers 0, 1, 2, 3, and so on, but that can only be proved by moving up to more powerful systems.[7]

The first incompleteness theorem tells us that moving up to a more powerful system is never going to be a complete cure. Increasing a system's power will not allow us to reach a system in which there is no incompleteness left, so long as we insist on still meeting the conditions on the system that we have laid down. There will always be a G. We have incurable incompleteness.

This does not mean that we should worry. We can do enormous amounts with the foundations for arithmetic that we have. There are also systems for other parts of mathematics which do not suffer from incompleteness, because they fail the condition of being powerful, but which are still very useful in areas other than arithmetic. As an example, systems for geometry can be like that.

12.1.2 Complete confidence in foundations

The second incompleteness theorem shows that we cannot use a system which meets our conditions to prove its own consistency. And as we noted in section 11.4, there

---

[7] One of the most easily comprehensible results is Goodstein's theorem, which concerns sequences of numbers that are generated in a particular way. These sequences start by shooting off to huge numbers, but the theorem states that if we keep on generating new members of a sequence in the prescribed way, we shall eventually come down to zero. So this is an example of incompleteness that is not an artificially constructed sentence like G. It is, however, an example of curable incompleteness, because Goodstein's theorem can be proved in a stronger system.

is no obvious way to get complete reassurance from other systems. So does the second theorem tell us that whatever foundations we establish, we can never have complete confidence in their firmness? Should the fear of inconsistency haunt us?

What the second theorem tells us is that we should not expect to obtain complete confidence from within systems of arithmetic. But this does not mean that we cannot obtain confidence from other sources. As noted in section 11.5, we can make general remarks that should help us to sleep at night. But there are also some more precise results. In particular, in 1936, Gerhard Gentzen (1909 − 1945) established the consistency of the standard system of arithmetic, by using methods that lie outside the arithmetic of finite numbers. This is an example of the use of a system that is weaker in some respects and stronger in others, as mentioned in section 11.4.4. Nervous people would say that this just transfers our worry to the other system that is used to prove consistency. Confident people would say that if we find support for some parts of mathematics in other parts of mathematics, and we never find a specific reason to worry about any of the parts, we should have confidence in all of the parts.

12.2   Provability and truth

We have so far avoided saying that sentences are true. We have not even said that simple sentences like "$8 + 7 = 15$" are true. But in everyday speech, it is normal to claim that mathematical statements are true. It is now time to say something about the relationship between provability and truth. This will tell us something about the significance of the first incompleteness theorem.

12.2.1 Systems and their interpretation

When we set up a system, we create a game that is played with symbols. Certain strings of symbols, the formulae, can be written. Some of those formulae are called sentences. And certain sequences of formulae are called proofs. But we could play the game without thinking about what it was supposed to achieve. We could just move symbols around, according to the rules, without any intention to say things about the world outside the game, the world in which there are numbers that we use to count things and that have certain properties. If we simply think in terms of a game, we should not really speak of numbers at all. Rather, we should speak of numerals, meaning ink marks on a page like "0", "1", "2", "3", and so on.

We do not, however, create systems merely in order to play games with symbols. We have created the standard system that we use for numbers in order to represent the numbers which obey rules that make those numbers remarkably useful in coping with the world.

Looked at another way, the system that we standardly use for numbers can be interpreted as being about the numbers 0, 1, 2, 3, and so on, and if we interpret it like that, we do not run into any practical difficulties. The evidence that we have got our system right is that when we see what we can prove, we find that we can prove sentences which, interpreted as being about those numbers, make claims that we find to be true – for example, the claim that when Winnie the Pooh puts eight pots of honey on a table, and then another seven pots, he has fifteen pots. If we had found that when we interpreted our chosen system in this way, we had been able to prove a sentence which would have led us to expect only fourteen pots of honey, we would have changed our system. A system which got things wrong like that would not be at all useful.

It may seem odd to talk about interpreting a system. We grasp the meaning of the sentence "$8 + 7 = 15$" immediately, without any feeling that there is a stage of interpreting it, like the stage of interpreting a sentence in some language that we are just beginning to learn. It would also be odd to talk about interpretation if each system had only one possible interpretation. But in fact many systems, including the system that we use for the numbers 0, 1, 2, 3, and so on, do have alternative interpretations – although the alternatives can be quite weird, and in everyday life we can safely ignore them. So it is worth identifying the interpretation of a system as a separate step.[8]

The important point for us is that provability and truth are separate things. We can generate proofs by playing games with symbols. But we need to interpret sentences, if we want to see whether they state truths about some subject matter or other.

12.2.2 The status of G

Now we can turn to what this tells us about the first incompleteness theorem. It is commonly said that G is true but unprovable. Should we say that?

---

[8] Not just any random interpretation of symbols is allowed. An interpretation has to fit the axioms. Take for example one of the axioms in the system that we use for the numbers 0, 1, 2, 3, and so on, the axiom "0 is not the successor of any number". An interpretation must respect this axiom. So we cannot interpret the system as applying to all of the integers, positive, zero and negative, because then every number would be the successor of a number. There would be no start of the chain of numbers, so there would be no number that we could call "0", in a way that would comply with the axiom. But even though interpretations must respect the axioms, there are some rather strange-looking alternative interpretations of the system that we use for the numbers 0, 1, 2, 3, and so on.

We shall consider provability first. This is provability within a system. There is no such thing as provability in absolute terms, because we give proofs within systems and we must follow the rules of whichever system we use.

So take a system that meets the conditions of identifiability of formulae, sentences, axioms and proofs, and of power. Consider its sentence G. The first incompleteness theorem does not entitle us to say whether G is provable within the system. What it does entitle us to say is that if the system is consistent, then G is not provable within the system, but if it is inconsistent, then G is provable within the system (along with every other sentence that we could write within the system).

Now suppose that we look at the system from the outside. We also note that if the system is consistent, there is indeed no number that is the Gödel number of a proof of G within the system, and that this is what G says. So we can say that if G is not provable within the system, it says something true. But we must remember the condition, "if G is not provable within the system", and the important words "within the system". It would be too sweeping to say that G said something true while being unprovable, without adding these qualifications.

12.2.3   First and second order systems

There is one more refinement to the picture painted here. Not only is provability relative to a system. Truth is also relative to interpretation. A sentence may say something true if interpreted in one way, but something false if interpreted in another way. We have already hinted at this, but it is worth reflecting on the point.

Logicians refer to interpretations of systems as **models**. A model of a system must respect the system's axioms.

But as we have already noted, there can be more than one way to respect a system's axioms. So a sentence might say something true in one model, and something false in another.

Whether there is more than one model, allowing for this possibility, depends on the extent to which the system allows generalization. We noted in section 7.3.2 that it is very useful to be able to generalize about all numbers. Systems that can be interpreted as allowing generalization over objects, but not over sets of objects, are called **first-order systems**. Such systems can easily have more than one model. Systems that can be interpreted as allowing generalization not only over objects, but also over sets of objects (for example, generalization over sets of numbers), are called **second-order systems**. Such systems can easily have just one model.

When a system can have more than one model, incompleteness is unsurprising. A sentence might be true in one model and false in another. Then there had better not be a proof of the sentence (because it would be false in the second model), and there had better not be a proof of its negation either (because that would be false in the first model).

In second-order systems, the problem is a bit different. There may be only one model, but there is no completely respectable notion of provability that can make provable all of the sentences which are true in the model. Truth outruns provability.

To return to first-order systems, where does this leave a claim that G says something true? It is a useful reminder that we must ask whether such a claim is a claim that there is a model of a system in which G for some system is true, or whether it is an informal claim that G says something true, made without reference to models of systems.

12.3 Misinterpretations of the theorems

Gödel's incompleteness theorems are remarkably open to misinterpretation. They are thought to imply conclusions that do not in fact follow. (It is a separate question whether the conclusions are correct. The point here is that they do not gain the support that they would gain, if they were consequences of Gödel's theorems.)

Some mistaken claims to the effect that certain conclusions follow appear to result from a little understandable carelessness in interpreting the theorems. Others can only be explained as results of cluelessness. There are degrees of misunderstanding between these extremes. We shall move along the scale, from understandable claims to bizarre ones.

The first mistaken claim to note lies within mathematics itself. It is the claim that there are some sentences such that neither they, nor their negations, can be proved in any system at all. This ignores the possibility of moving up from a system in which some sentence is not provable, to another one in which it is provable. "In any system, there is a sentence such that neither it nor its negation can be proved", is not the same as "There is a sentence, such that neither it nor its negation can be proved in any system". If the sentence is the G of some system, call it System 1, then we move up to System 2, we need to be explicit that the G in question says that it is not provable in System 1. But we can do that.[9]

A related mistaken claim is that the consistency of a system can never be proved. It may not be possible to

---

[9] We do not do it by reproducing G symbol for symbol and then inserting "(in System 1)". Instead, we prove in System 2 that if System 1 meets the conditions, there is no number that is the Gödel number of a proof in System 1 of the diagonalization of the almost-sentence with Gödel number 123,456. (Remember that we are using 123,456 as a made-up Gödel number for the almost-sentence F.)

prove the consistency of a consistent system within that system, but we might be able to prove it within another system – although we would then have to worry about the consistency of the second system.

Moving on to physics, one mistaken claim is that the theorems show that we could never work out a set of physical laws that would completely govern the Universe. The idea is that for any given set of laws, there might be something going on that was not captured by those laws. The mistake here is to assume that the physical Universe is a formal system of the type that we have defined, and that it meets the conditions we laid down for incompleteness to follow. Maybe the physical Universe is like that, but we would have to demonstrate that supposed fact. We should not just assume it.

Turning to our minds, one mistaken claim is that the theorems show that we can prove mathematical results which computers could not prove. People who think this are likely to start from the correct observation that human beings can have insights which computers do not yet have. But insight is not the same thing as proof. Proof takes place within a formal system, and there is no reason to suppose that our brains are better at handling formal systems than computers will be, once developments which are already reasonably foreseeable have taken place. There is not even any reason to suppose that we shall always stay ahead of computers in powers of creative insight.

Finally, there are claims that are based on the massive error of supposing that the theorems are about the world, our knowledge and our processes of thought in general, rather than their being about formal systems as defined. People claim that the theorems show that the world is indeterministic, that we have free will, that any knowledge we may claim to have is relative to a language which

must itself be interrogated, that no legal code can ever be comprehensive, and that there is room for a god to have played a role in the origin of the Universe.

One point should be conceded to some of those who appear to misunderstand Gödel's work. Some people only use his work as a source of inspiration, analogy or metaphor. There is nothing wrong with making that kind of use of the incompleteness theorems, just as there is nothing wrong with allowing works of art to trigger philosophical ideas. But such use gives no demonstration at all of the worth of the ideas that result. Ideas do not derive any support from merely psychological connections to mathematical theorems. Those who make such connections prove nothing by making them. Gödel, on the other hand, proved a very great deal.

13 Further reading

We have avoided mathematical symbols and proofs. But if you want to get to grips with Gödel's work in any detail, you will need to read mathematical formulations of it. Some of the suggestions here are distinctly mathematical.

Books written in this mathematical way conduct the argument within the logical systems under investigation, rather than by talking about systems from the outside. That is, they use the method of arithmetization that we discussed in section 8.3.

Different authors take different routes to the theorems. Some authors like to emphasize symbols, regarded as mere marks on the page with no meanings. Others prefer to interpret the symbols as referring to numbers, regarded as objects that exist outside systems. If an author seems to be doing something different from what we have done here,

the most likely reason is that he or she is using a different route. The different routes are equally good, although they do tend to bring out some different points.

13.1 Internet resources

Panu Raatikainen has written an article for the Stanford Encyclopedia of Philosophy at:

http://plato.stanford.edu/entries/goedel-incompleteness/

Peter Smith has supplied some free notes at:

http://www.logicmatters.net/igt/godel-without-tears/

13.2 Books

There is an outstandingly clear and thorough treatment of Gödel's theorems, and of several related topics, in Peter Smith, *An Introduction to Gödel's Theorems* (second edition, Cambridge, Cambridge University Press, 2013). Be sure to get the second edition, published in 2013, not the first edition, published in 2007, because the author made significant improvements in the second edition.

Supporting materials for Peter Smith's book can be found at:

http://www.logicmatters.net/igt/

Another treatment of the theorems, which is rather different in style, and which includes chapters on philosophical questions that are more or less closely linked to the theorems, is Francesco Berto, *There's Something About Gödel* (Oxford, Blackwell, 2009).

A very good book on the significance of the theorems is Torkel Franzén, *Gödel's Theorem: An Incomplete Guide to Its Use and Abuse* (Wellesley, MA, A K Peters, 2005). The title uses the singular, "theorem", but the author meant this to cover both theorems.

There is a detailed treatment of techniques that generate the sentence G for a system, and of many other topics, more or less related to those techniques, in Raymond Smullyan, *Diagonalization and Self-Reference* (Oxford, Clarendon Press, 1994). This book is now printed on demand, and it is expensive, but it may be available secondhand or in libraries.

There is a chapter on Gödel in Robert S. Wolf, *A Tour Through Mathematical Logic* (Washington, DC, Mathematical Association of America, 2005). This book gives a good survey of the whole field of mathematical logic.

14   List of changes

14.1   The current version: version 4.3

This is version 4.3 of the paper, dated 2 October 2016. The one textual change from version 4.2 is that the author's contact details have been changed to refer to a contact page on his website.

14.2   Previous versions

14.2.1   Version 1

Version 1 was dated 28 May 2014.

14.2.2 Version 2

Version 2 was dated 24 December 2014. Changes from version 1 were as follows.

The style of links was changed, so that they appeared as text in blue rather than as text enclosed in red boxes.

More pagebreaks were included, so as not to end pages just after the starts of sections or in other inconvenient places. There were therefore more blank spaces at the ends of pages.

In section 3.2, the proof that the Earth is round was replaced by a proof that smoking is not allowed in the dining room of a restaurant. (Peter Smith kindly pointed out that the old proof, which relied on photographs showing that the Earth was round, was either largely redundant or invalid, depending on the sense of "show".)

In section 10.2.2, footnote 3 was amended so as to be better aligned with the main text. It was changed to discuss ω-inconsistency instead of ω-consistency.

The text in section 12.2 was changed to align the example of pots of honey with the start of chapter 3 of *The House at Pooh Corner*. The relevant sum became $8 + 7 = 15$.

In section 12.2, the end of the first paragraph was amended to make it clear that the material in that section tells us about the significance of the first incompleteness theorem, rather than of both theorems.

In section 12.2, footnote 8 was re-worded slightly so as to keep it on one page.

14.2.3 Version 3

Version 3 was dated 4 February 2015. Changes from version 2 were as follows.

A new section 8.3, on arithmetization, was inserted. There were related changes to the text of the rest of section 8. The numbering of later sections was not affected. There were also minor consequential changes to the text of section 3.2.

A note on arithmetization was added to section 13.

The way to state consistency in section 11.3 was amended to allow any provable sentence to be used.

There were several minor changes to wording throughout the paper.

14.2.4   Version 4.0

Version 4.0 of the paper was dated 17 November 2015. Changes from version 3 were as follows.

Section 12.2 was revised and expanded to say something about models and about second-order systems.

There were some minor textual changes elsewhere.

The layout was changed to reduce the number of characters per line, for the sake of readability. Page numbers therefore changed.

14.2.5   Version 4.1

Version 4.1 of the paper was dated 2 February 2016. The one change from version 4.0 was a re-wording of the last paragraph of section 11.1.

14.2.6   Version 4.2

Version 4.2 of the paper was dated 16 September 2016. The one textual change from version 4.1 was that section

1.2 was amended to refer to a Creative Commons licence. Pagination also changed slightly, largely in order to keep some long footnotes on single pages. The usefulness of doing so was thought to be worth the consequent introduction of a few large white spaces.

END OF PAPER