Gödel Without (Too Many) Tears

Kurt Gödel's famous First Incompleteness Theorem shows that for any sufficiently rich theory that contains enough arithmetic, there are some arithmetical truths the theory cannot prove. How is this remarkable result proved? This short book explains. It then also discusses Gödel's Second Incompleteness Theorem. Based on lecture notes for a short course given in Cambridge for many years, the aim is to make the theorems available, clearly and accessibly, even to those with a limited formal background.

Peter Smith was formerly Senior Lecturer in Philosophy at the University of Cambridge. His books include *Explaining Chaos* (1998), *An Introduction to Formal Logic* (2003; 2020) and *An Introduction to Gödel's Theorems* (2007; 2013). He was also editor of *Analysis* for a dozen years.

The page intentionally left blank

Gödel Without (Too Many) Tears

Second edition

Peter Smith

LOGIC MATTERS

Published by Logic Matters, Cambridge

A low-cost paperback of this book is available by print on demand from Amazon

Additional resources for this publication at logicmatters.net/igt.

Internal references within the PDF are live links.

Since this book is not produced by a publisher with a marketing department, your university librarian will not get to know about it in the usual way. You will therefore need to give them the details and ask them to order a printed copy for the library.

Contents

The page intentionally left blank

Preface

Why this short book? After all, I have already written a rather long book, *An Introduction to Gödel's Theorems*, originally published by CUP, now freely downloadable. Surely that's more than enough to be going on with?

Ah, but there's the snag. It *is* more than enough. In the writing, as is the way with these things, that book grew far beyond the scope of the original notes on which it was based. And while I hope the result is still quite accessible if you are prepared to put in the required time and effort, there is – to be frank – a *lot* more material in the book than is really needed by those wanting a first encounter with the famous incompleteness theorems.

Quite a few readers might therefore appreciate a cut-down version of some of that material – an introduction to the *Introduction*, if you like. Hence *Gödel Without (Too Many) Tears*. There are occasional footnotes referring to sections of the longer book, indicating where topics are discussed further: but you don't have to chase up those references to get a more limited but still coherent story in this shorter version.

There isn't much purely philosophical discussion here in *GWT*. The aim, rather, is to put you in a position where you have a secure enough understanding of enough of what's going on logically that you can sensibly make a start on thinking about any (supposed) philosophical implications.

So what background do I presuppose? What do you need to bring to the party? Very little. If you have a grasp of a modest amount of elementary logic, and have the patience to follow some simple mathematical arguments, you should have little difficulty in following the exposition here. I have given proofs of most of the important theorems I state, especially if the proofs involve some neat ideas. But I have left a few proofs for enthusiasts to follow up elsewhere, when trekking through the details has little intrinsic interest.

GWT started life as a set of notes written to accompany the last outings of a short lecture course given in Cambridge (which was also repeated at the University of Canterbury, NZ). The notes aimed to bridge the gap between my classroom talk'n'chalk which just highlighted the Really Big Ideas, and the much more detailed treatments of topics available in *IGT*. However, despite that intended role, I did try to make the notes reasonably stand-alone.

Those notes were tied to the first edition of *IGT*, as published in 2007. A significantly improved second edition of the book, *IGT2*, was published in 2013,

which prompted me to revise the notes. Then came the pandemic in 2020; rewriting the notes again and turning them into a book became occupational therapy to distract me a little from the world's manifold troubles. The result was the first edition of *GWT* in book form.

This new version corrects known errors in the first edition, adds a short new chapter, and makes a lot of small stylistic improvements, enough revisions to make it more than just a corrected reprint. So although the changes aren't radical, let's count it as a new edition.

Many thanks to Henning Makholm for comments on the original notes for *GWT*, and also to David Auerbach, Sam Butchart, David Furcy, David Makinson and Rowsety Moid for more comments that helped shape the resulting book. I should also thank Ben Selfridge for pointing out the most serious glitch in the first edition, that prompted me to get to work on this edition. But many others too have at various stages kindly let me know about typos and more serious mistakes, or made helpful suggestions. *More thanks to be added.* I really am very grateful to everyone!

1 A very brief note on Kurt Gödel

By common agreement, Kurt Gödel (1906–1978) was the greatest logician of the twentieth century.

Born in what is now Brno, and educated in Vienna, Gödel left Austria for the USA in 1940, and spent the rest of his life at the Institute for Advanced Study at Princeton.

Gödel's doctoral dissertation, written when he was 23, established the *completeness* theorem for the predicate calculus (showing for the first time that a standard proof system for first-order logic does indeed capture all the semantically valid inferences).

Later he would do immensely important work on set theory, as well as make seminal contributions to proof theory and to the philosophy of mathematics. He even wrote about models of General Relativity with 'closed timelike curves' (where, in some sense, time travel is possible). But always a perfectionist, he became a very reluctant publisher: some of his philosophically most interesting work is in the substantial volume of Unpublished Essays and Lectures in his *Collected Works*.

Gödel proved a lot of important results, then. But talk of 'Gödel's Theorems' typically refers to the two *incompleteness* theorems he presented in an epoch-making 1931 paper. And it is these theorems, and more particularly the First Theorem, that this book is all about. (Yes, that's right: Gödel did prove a 'completeness theorem' and also 'incompleteness theorems'. I'll explain the difference very soon.)

The impact of the incompleteness theorems on foundational studies is hard to exaggerate. For a start, putting it crudely and a bit tendentiously, they sabotage the ambitions of two major programmes in the foundations of mathematics – logicism and Hilbert's Programme.

We'll say just a little about logicism in the next chapter, and something about Hilbert's Programme much later, when we get round to discussing the Second Theorem in Chapter **??**. But you don't have to know anything about this background to find the two theorems intrinsically fascinating. And as we will see, the beautiful ideas underlying their proofs are surprisingly easy to understand.

So now read on ...

2 Incompleteness, the very idea

The title of Gödel's great 1931 paper translates as *'On formally undecidable propositions of Principia Mathematica and related systems I'*.

The 'I' here indicates that this was intended to be the first part of a two part paper, with Part II spelling out in detail the proof of the Second Theorem which is only very briefly indicated in Part I. But Part II was never written. We'll see in due course why not.

This title itself gives us a number of things to explain. What's a 'formally undecidable proposition'? What is *Principia Mathematica*? Ok, you've probably heard of that triple-decker work by A. N. Whitehead and Bertrand Russell, more than a century old and now very little read except by historians of logic: but what is the project of that book? And what counts as a 'related system' – a system suitably related, that is, to the one in *Principia*? In fact, just what is meant by 'system' here?

Let's take the last question first. We will take a 'system' (in the relevant sense) to be an *effectively axiomatized formal theory*.[1] But what does that mean?

2.1 The idea of an effectively axiomatized formal theory

The general idea of an axiomatized theory is no doubt familiar. But now we need to be more specific: our focus is going to be on theories which, in headline terms, have

(i) an effectively formalized language,
(ii) an effectively decidable set of axioms, and
(iii) an effectively formalized proof system.

We'll explain these headlines in just a moment. First, though, the new idea you need to get your head round here is the intuitive notion of *effective decidability*.

Let's say, as a first shot:

Defn. 1. *A property P (defined over some domain of objects D) is* effectively decidable *iff*[2] *there's an algorithm (a finite set of instructions for a deterministic*

[1] It will turn out that Gödel originally had in mind just a central subclass of systems in this wide sense; but let's not complicate the story yet.

[2] 'Iff' is of course the logician's abbreviation for 'if and only if'.

computation) for settling in a finite number of steps, for any object $o \in D$, whether o has property P.

To put it another way, a property is effectively[3] decidable just when there's a step-by-step mechanical routine for settling whether o has property P, such that a suitably programmed deterministic computer could in principle implement the routine (idealizing away from practical constraints of time, etc.).

Two easy and familiar examples from propositional logic: the property of being a tautology is effectively decidable (by a truth-table test); so is the property of being the main connective of a sentence (mainly by bracket counting).

How satisfactory is our first-shot definition, though? To elucidate it, we appealed to the idea of what an idealized computer could in principle do by implementing some algorithmic procedure. This idea plainly stands in need of further elaboration. It turns out, however, that the notion of effective decidability is very robust: what is algorithmically-computable-in-principle according to one sensible sharpened-up definition is exactly the same as what is algorithmically-computable-in-principle according to any other sensible sharpened-up definition. Of course, it's not at all obvious that this is how things are going to pan out. So for the moment you are going to have to take it on trust (sorry!) that Defn. 1 *can* call upon a determinate notion of algorithmic computability. Still, our current rough-and-ready explanations will suffice for present purposes, in clarifying conditions (i) to (iii) for being an effectively axiomatized formal theory.

(i) We'll assume that the basic idea of a *formalized language* L is reasonably familiar. But note that a language, for us, has both a *syntax* and an intended *semantics*:

(1) The syntactic rules fix which strings of symbols form terms, which form wffs (i.e. well-formed formulas), and in particular which strings of symbols form sentences, i.e. closed wffs with no unbound variables dangling free.

(2) The semantic rules assign interpretations, i.e. assignments of truth-conditions, to every sentence of the language.

It is not at all unusual for logicians to call a system of uninterpreted strings of symbols a 'language'. But I really think we should deprecate that usage. Sometimes below I'll talk about an 'interpreted' language for emphasis: but strictly speaking, by my lights, that's redundant.

The familiar way of presenting the syntax of a formal language is by first specifying some finite[4] set of basic logical and non-logical symbols, and then giving rules for building up more and more complex expressions from these symbols. This is done in such a way that there are effective algorithmic procedures for deciding e.g. whether a given string of symbols counts as a term, or a wff, or a

[3] It is in fact common to talk just about 'decidability'. But here at the outset it is probably helpful if I keep adding 'effectively' for emphasis.

[4] "Finite? But might we not need an unlimited, potentially infinite, supply of variables, for example?" Sure. But we can build up an infinite list of variables from finite resources, as in '$\mathsf{x}, \mathsf{x}', \mathsf{x}'', \mathsf{x}''', \ldots$'. We lose no relevant generality in keeping our basic symbol-set finite.

wff with one free variable, or a sentence; and there will be an effective procedure too for recovering from a sentence its 'constructional history', tracing the unique way it can be syntactically built up from its ultimate symbolic constituents.

The familiar way of presenting the semantics is then to assign semantic values to the basic non-logical expressions of the language and fix domains of quantification, and to give rules for effectively working out the truth-conditions of sentences in terms of the unique way they are syntactically built up from their parts. (Do read that carefully. What we should be able to mechanically work out is what the sentence *says*. But it is of course one thing to work out the conditions under which a sentence is true, and – usually – something quite different to work out whether those conditions are met, i.e. work out whether the sentence actually *is true*!)

So let's wrap that up in summary form:

Defn. 2. *An interpreted language L is* effectively formalized *iff (a) it has a finite set of basic symbols, (b) syntactic properties such as being a term of the language, being a wff, being a wff with one free variable, and being a sentence, are all effectively decidable and the syntactic structure of any sentence is effectively determinable, and (c) this syntactic structure together with L's semantic rules can be used to effectively determine the unique intended interpretation of any sentence.*

Now, *why* do we want (b) the syntactic properties of being a sentence, etc., to be effectively decidable? Well, the very point of setting up a formal language is, for a start, to put issues of what is and what isn't a well-formed sentence beyond dispute, and the best way of doing that is to ensure that even a suitably programmed computer could decide whether a string of symbols is or is not a sentence of the language.

And *why* do we want (c) the unique truth-conditions of a sentence to be effectively determinable? Because we don't want any ambiguities or disputes about interpretation either.

(ii) A theory is sometimes defined to be just any old set of sentences. We are concerned, though, with the more structured notion of an *axiomatized theory*. In this case, we pick out some bunch of sentences Σ as giving *axioms* for the theory T; we also give T some *proof system*, i.e. some deductive apparatus; and then all the sentences that are derivable from axioms in Σ using the deductive apparatus are T's *theorems*.

But what does it take for T to be an *effectively* axiomatized formal theory, apart from the obvious condition that it uses an effectively formalized language? For a start, we require it to have an effectively decidable set of axioms, meaning that the property of being a T-axiom is effectively decidable. Why? Because if we are in the business of pinning down a theory by axiomatizing it, then we will normally want to avoid any possible dispute about what counts as a legitimate starting point for a proof by ensuring that we can mechanically decide whether a given sentence really is one of the axioms.

(iii) But just laying down a bunch of axioms would be pretty idle if we can't deduce conclusions from them! An axiomatized theory T will, as we said, come equipped with a proof system, a set of logical rules for deriving further theorems from our initial axioms. But a proof system such that we couldn't routinely tell whether its rules are being followed again wouldn't have much point for practical purposes. Hence it is natural to require that T's logic has an effectively formalized proof system, i.e. one where it is effectively decidable whether a given array of wffs is a well-constructed derivation from the axioms according to the rules of the proof system. It doesn't matter for our purposes, though, whether the proof system is an axiomatic logic, a natural deduction system, a tree/tableau system, or a sequent calculus – so long as it is effectively checkable that a candidate proof-array has the property of being properly constructed according to the rules of the proof system.

Careful again, though! To say that it must be effectively decidable whether a candidate T-proof of φ is a kosher proof is not, repeat *not*, to say that it must be effectively decidable whether φ actually *has* a T-proof. To stress the point: it is one thing to be able to effectively *check* that some proposed proof follows the rules; it is another thing to be able to effectively *decide in advance* whether there exists a proof waiting to be discovered. (Looking ahead, we will see as early as Chapter 5 that any formal effectively axiomatized theory T containing a modicum of arithmetic is such that, although you can mechanically check a purported proof of φ to see whether it *is* a proof, there's no mechanical way of telling of an arbitrary φ whether it is provable in T or not.)

So, in summary of (i) to (iii),

Defn. 3. *An* effectively axiomatized *formal theory T has an effectively formalized language L, a certain class of L-wffs are picked out as axioms where it is effectively decidable what's an axiom, and it has a proof system such that it is effectively decidable whether a given array of wffs is a derivation from the axioms according to the rules.*

From now on, when we talk about formal theories, we will be concerned with effectively axiomatized formal theories (unless we explicitly say otherwise).

2.2 A quick reminder about logical proof systems

Let's have some more familiar logical notation. Suppose S is a logical proof system:

Defn. 4. *'$\Sigma \vdash_S \varphi$' says that there is a formal derivation in the proof system S from sentences in Σ to the sentence φ as conclusion.*

'$\Sigma \models \varphi$' says that Σ logically entails φ, i.e. any way of (re)interpreting the relevant non-logical vocabulary that makes all the sentences in Σ true makes φ true too.

So '$\vdash_S$' signifies deducibility in S, which is a *syntactically* defined relation (being a well-formed proof is a question of being of the right symbolic shape, which is

determined by syntactic pattern-matching). By contrast, '$\vDash$' signifies a *semantically* defined relation.

Of course, we normally want a formal deduction to be truth-preserving; so we will want our proof system S to respect logical entailments, requiring that if $\Sigma \vdash_S \varphi$ then indeed $\Sigma \vDash \varphi$. In a word, we require an acceptable logical proof system to be *sound*.

We can't in general insist on the converse. Not every relation of logical entailment can be captured in a proof system S for which it is effectively decidable what counts as an S-proof. But take the very important special case where we are working in a classical first-order setting, so the relevant logical vocabulary is just the truth-functional propositional connectives, the identity predicate, plus the apparatus of quantification. In this case, if Σ logically entails φ, then there will indeed be a formal deduction of φ from those sentences in your favourite first-order logical system S: i.e. if $\Sigma \vDash \varphi$ then $\Sigma \vdash_S \varphi$. In a word, there can be a *complete* deductive proof system S for first-order logic. As noted before, this was first shown for the particular case of a Hilbert-style axiomatic deductive system by Gödel in his 1923 doctoral thesis: hence *Gödel's completeness theorem*.

2.3 'Formally undecidable propositions' and negation incompleteness

We will recycle the familiar notation, for application to a formal theory T:

Defn. 5. *'$T \vdash \varphi$' says that there is a formal derivation in T's proof system from T's axioms to the sentence φ as conclusion (in short, φ is a T-theorem).*

Now, we will be interested in what claims a theory T can settle, one way or the other. So, assuming '$\neg$' is T's negation sign, we say

Defn. 6. *If T is a theory, and φ is some sentence of the language of that theory, then T* formally decides *φ iff either $T \vdash \varphi$ or $T \vdash \neg\varphi$. Hence, a sentence φ is* formally undecidable *by T iff $T \nvdash \varphi$ and $T \nvdash \neg\varphi$.*

A related bit of terminology:

Defn. 7. *A theory T is* negation complete *iff it formally decides every closed wff of its language – i.e. for every sentence φ, $T \vdash \varphi$ or $T \vdash \neg\varphi$.*

So there are formally undecidable propositions in a theory T if and only if T isn't negation complete.

It might help to fix ideas, and distinguish the two notions of completeness – semantic completeness for a system of logic, negation completeness for a theory – if we look at a toy example.

Suppose then that theory T is built in a propositional language with just three propositional atoms, $\mathsf{p}, \mathsf{q}, \mathsf{r}$, plus the usual propositional connectives. We give T a standard propositional classical logic (pick your favourite flavour of system!). And assign T just a single non-logical axiom: $(\mathsf{p} \wedge \neg\mathsf{r})$.

Then, just by assumption, T has a *semantically-complete logic*, since standard propositional calculi are complete. Hence, for any wff φ of T's limited language,

if $T \vDash \varphi$, i.e. if T tautologically entails φ, then $T \vdash \varphi$.

However, trivially, T is not a *negation-complete theory*. For example T can't decide whether q is true. And there are lots of other wffs φ for which both $T \nvdash \varphi$ and $T \nvdash \neg\varphi$.

Our toy example shows that it is very, very easy to construct negation-incomplete theories with formally undecidable propositions: just hobble your theory T by leaving out some key assumptions about the matter in hand!

On the other hand, suppose we are trying to fully pin down some body of truths (e.g. the truths of basic arithmetic) using a formal theory T. We fix on an interpreted formal language L apt for expressing such truths. Then we'd ideally like to lay down enough axioms framed in L to give us a theory T such that, for any L-sentence φ, if φ is true then $T \vdash \varphi$. So, making the classical assumption that either φ is true or $\neg\varphi$ is true, we'd very much like T to be such that for any φ, either $T \vdash \varphi$ or $T \vdash \neg\varphi$ (but, of course, not both).

In other words, it is very natural to aim for theories T which are indeed negation complete.

2.4 Seeking a negation-complete theory of arithmetic

The elementary arithmetic of addition and multiplication is child's play (literally!). So surely we should be able to wrap it up in a nice formal theory, aiming for negation completeness.

Let's first fix on a formal *language of basic arithmetic* designed to express elementary arithmetical propositions. We will give this language

(i) a term '0' to denote zero; and
(ii) a sign 'S' for the successor (i.e. 'next number') function.

This means that we can construct the sequence of terms '0', 'S0', 'SS0', 'SSS0', ... to denote the natural numbers 0, 1, 2, 3, ... These are our language's *standard numerals*, and by using a standard numeral our language can denote any particular natural number.

We will also give this language

(iii) function signs for addition and multiplication, together with
(iv) the usual first-order logical apparatus including the identity sign, where
(v) quantifiers are interpreted as running over the natural numbers.

(We aren't building in subtraction and division as primitives. But subtraction is definable in terms of addition, formalizing the idea that $n - m$ is the number k such that $m + k = n$, if there is such a number. And similarly division is definable in terms of multiplication.)

Now, it is entirely plausible to suppose that, whether or not the answers are readily available to us, questions posed in this language of basic arithmetic have entirely determinate answers. Why? Well, take the following two bits of data:

(a) The fundamental zero-and-its-successors structure of the natural number series.
(b) The nature of addition and multiplication as given by the school-room explanations.

By (a) we mean that zero is not a successor, every number has a successor, distinct numbers have distinct successors, and so the sequence of zero and its successors never circles round but marches on for ever: moreover there are no strays – i.e. every natural number is in that sequence starting from zero. By (b) we mean to cover such basic laws as that $m + n = n + m$ – we will say more about this in due course. It is very plausible to suppose that facts of the kind (a) and (b) together should fix the truth-value of every sentence of the language of basic arithmetic – after all, what more could it take?

But (a) and (b) seem so very basic and straightforward. So we will surely expect to be able to set down some axioms which (a) characterize the number series, and (b) define addition and multiplication: in other words, we should surely be able to frame axioms which codify what we teach the kids. And then the thought that (a) and (b) fix the truths of basic arithmetic becomes the thought that our axioms capturing (a) and (b) should settle every such truth. In other words, if φ is a true sentence of the language of successor, addition, and multiplication, then φ is provable from our axioms (and if φ is a false sentence, then $\neg\varphi$ is provable).

In sum, whatever might be the case with fancier realms of mathematics, it is very natural to suppose that we should at least be able to set down a negation-complete (and effectively axiomatized) formal theory of basic arithmetic.

2.5 Logicism and *Principia*

Now let's pause at this point to bring *Principia* into the story.

It is natural to ask: what is the *status* of the axioms of a formal theory of basic arithmetic? For example, what is the status of the formalized version of a truth like 'every number has a unique successor'? That hardly looks like a mere empirical generalization (something that could in principle be empirically refuted).

I suppose you might be a Kantian who holds that the axioms encapsulate 'intuitions' in which we grasp the fundamental structure of the numbers and the nature of addition and multiplication, where these 'intuitions' are a special cognitive achievement in which we somehow represent to ourselves an abstract arithmetical world.

But talk of such intuitions is, to say the least, puzzling and problematic. So we could very well be tempted instead by Gottlob Frege's seemingly more straightforward view that the axioms of arithmetic are *analytic*, simply truths of logic-plus-definitions. On this view, we don't need Kantian 'intuitions' going beyond logic: logical reasoning from mere definitions is enough to get us the axioms of arithmetic, and more logic gives us the rest of the arithmetic truths

from these axioms. And hopefully the fundamental definitions of arithmetical primitives like 'one' need involve no more than logical ideas (after all, remember how we can express 'there is exactly one F' using just logical notation). This Fregean line – that arithmetic can be grounded in logic-plus-definitions – is standardly dubbed *logicism*.

If this proposal is to be more than wishful thinking, we need a well-worked-out logical system within which to pursue a logicist derivation of arithmetic. Famously, and to his eternal credit, Frege gave us the first competent system of quantificational logic in his *Begriffsschrift* of 1879. But equally famously, Frege's own attempt to go on to be a logicist about basic arithmetic (in fact, for him, about significantly more than basic arithmetic) hit the rocks, because – as Russell showed – the full deductive proof system that he later used, going beyond core quantificational logic, is inconsistent in a pretty elementary way. Frege's full system is beset by Russell's Paradox.[5]

That disaster devastated Frege; but Russell himself was undaunted. Still gripped by logicist ambitions he wrote:

> All mathematics [yes! – *all* mathematics] deals exclusively with concepts definable in terms of a very small number of logical concepts, and ... all its propositions are deducible from a very small number of fundamental logical principles.

That's a huge promissory note in Russell's *The Principles of Mathematics* (1903). And *Principia Mathematica* (three volumes, though unfinished, 1910, 1912, 1913) is Russell's attempt with Whitehead to start making good on that promise.

The project of *Principia*, then, is to set down some logical axioms and definitions from which we can deduce, for a start, all the truths of basic arithmetic (so giving us a negation-complete theory at least of arithmetic). Famously, the authors eventually get to prove that $1 + 1 = 2$ at *110.643 (Volume II, page 86), accompanied by the wry comment, 'The above proposition is occasionally useful'. So far so good! But can Russell and Whitehead, in principle, prove *every* truth of arithmetic?

2.6 Gödel's bombshell

Principia, frankly, is a bit of a mess – in terms of clarity and rigour, it's quite a step backwards from Frege's logical systems. There are technical complications, and not all *Principia*'s axioms are clearly 'logical' even in a stretched sense. In particular, there's an appeal to a brute-force *Axiom of Infinity* which in effect stipulates that there is an infinite number of objects. But we don't need to go into details; for we can leave such worries aside – they pale into insignificance compared with the bombshell exploded by Gödel.

[5] Roughly, Frege's full system implies that there is a set of all sets which are not members of themselves – but ask: does that set belong to itself?

2 Incompleteness, the very idea

For Gödel's First Incompleteness Theorem sabotages not just the grand project of *Principia* but – as advertised in the title of his paper – shows that *any* similar attempt to pin down *all* the truths of basic arithmetic in a theory with nice properties like being effectively axiomatized is in fatal trouble. His First Theorem says – at a rough first shot – that *nice theories containing enough arithmetic are always negation incomplete*. So given any nice effectively axiomatized formal theory T, there will be arithmetic truths that can't be proved in that particular theory.

Only a moment ago, it didn't seem at all ambitious to try to capture all the truths of basic arithmetic in a single (consistent, effectively axiomatized) theory. But attempts to do so – and in particular, attempts to do this in a way that would appeal to Frege and Russell's logicist instincts – must always fail. Which is a rather stunning result![6]

How did Gödel prove his result? Well, let's pause for breath; the next chapter explains more carefully what the theorem (in two versions) claims, and then in Chapter 4 we outline a Gödelian proof of one version.

[6] 'Hold on! I've heard of 'neo-logicism' which has its enthusiastic advocates. How can that be so if Gödel showed that logicism is a dead duck?'

Well, we might still like the idea that some logical principles plus what are more-or-less definitions (in a language richer than that of first-order logic) together *semantically* entail all arithmetical truths – even if we can't capture the relevant semantic entailment relation in a single effectively axiomatized deductive system of logic. Then the resulting overall system of arithmetic won't count as a formal effectively axiomatizable theory; so Gödel's theorems won't straightforwardly apply. But all that is another story.

3 The First Theorem, two versions

3.1 Soundness, consistency, etc.

Let's read into the record two more, no doubt familiar, definitions:

Defn. 8. *A theory T is* sound *iff its axioms are true (on the interpretation built into T's language), and its proof system is truth-preserving, so all its theorems are true.*

Defn. 9. *A theory T is (syntactically)* consistent *iff there is no φ such that $T \vdash \varphi$ and $T \vdash \neg\varphi$, where '$\neg$' is T's negation operator.*

In a classical setting, if T is inconsistent, then $T \vdash \varphi$ for all φ. So another way of defining consistency is by saying that T is consistent iff for some φ, $T \nvdash \varphi$. And of course, soundness implies consistency. We shouldn't need to delay over these ideas.

But we also need another (quite natural) definition to use in this chapter:

Defn. 10. *The formalized interpreted language L* contains the language of basic arithmetic *iff L has a term which denotes zero and function symbols for the successor, addition and multiplication functions defined over numbers – these can be either built-in as primitives or introduced by definition – and has the usual connectives, the identity predicate, and can express quantifiers running over the natural numbers.*

An example might be the language of set theory, in which we can define zero, successor, addition and multiplication in standard ways, and express restricted quantifiers running over just zero and its successors.[1]

3.2 Two theorems distinguished

In his 1931 paper, Gödel proves (more or less) the following: [2]

[1] Is the system of numbers referred to in set theory the genuine article or just a structurally equivalent surrogate? We are not going to tangle with *that* messy issue! When we talk of quantifying over numbers inside e.g. set theory, then, understand that to be quantifying either over natural numbers or over whatever surrogates we can take to play the role of natural numbers there. Nothing relevant to our project hangs on the difference.

[2] I say 'more or less' because, as footnoted in §2.1, Gödel's initial idea of a formalized theory was in fact a bit narrower than our notion of an effectively axiomatized theory.

3 The First Theorem, two versions

Theorem 1. *Suppose T is an effectively axiomatized formal theory whose language contains the language of basic arithmetic. Then, if T is sound, there will be a true sentence G_T of basic arithmetic such that $T \nvdash \mathsf{G}_T$ and $T \nvdash \neg\mathsf{G}_T$, so T is negation incomplete.*

We will outline a pivotal part of Gödel's proof in the next chapter.

However this version of an incompleteness theorem *isn't* what is most commonly referred to as *the* First Theorem, nor is it the result that Gödel foregrounds in his 1931 paper. For note, Theorem 1 tells us what follows from a *semantic* assumption, namely the assumption that T is sound. And soundness is defined in terms of truth.

Now, post-Tarski, most of us aren't particularly scared of the notion of truth. To be sure, there are issues about how best to treat the notion formally, to preserve as many as possible of our pre-formal intuitions while e.g. blocking the Liar Paradox. But most of us don't regard the relevant notion of a sound theory as metaphysically loaded in an obscure and worrying way. However, Gödel was writing at a time when – for various reasons (think logical positivism!) – the very idea of truth-in-mathematics was under some suspicion. It was therefore *extremely* important to Gödel that he could show that we don't need to deploy any semantic notions to get an incompleteness result. So he goes on to demonstrate a result which we can put schematically like this (more or less):

Theorem 2. *Suppose T is an effectively axiomatized formal theory whose language contains the language of basic arithmetic. Then, if T is consistent and can prove a certain modest amount of arithmetic (and has an additional property that any sensible formalized arithmetic will share), there will be a sentence G_T of basic arithmetic such that $T \nvdash \mathsf{G}_T$ and $T \nvdash \neg\mathsf{G}_T$, so T is negation incomplete.*

Being consistent is a syntactic property; being able to formally prove enough arithmetic is another syntactic property; and the mysterious additional property which I haven't explained is syntactically defined too. So *this* version of the incompleteness theorem only makes syntactic assumptions.

Of course, we'll need to be a lot more explicit about the details in due course; but this indicates the general character of Gödel's result in the second version. Our 'can prove a certain modest amount of arithmetic' gestures at what it takes for a theory to be sufficiently related to *Principia*'s for the theorem to apply (recall the title of the 1931 paper). But I'll not pause here to spell out just how much arithmetic that is, though we'll eventually find that it is stunningly little.[3]

For now, then, the first key take-away message of this chapter is that the incompleteness theorem does come in two different flavours. There's a version making a *semantic* assumption (the relevant theory T needs to be expressively rich enough and sound), and there's a version making only *syntactic* assumptions (about what T can and can't derive from its axioms). It is important to keep this firmly in mind.

[3] Nor will I pause to explain that 'additional property' condition. We'll meet it in due course, but also eventually see how – by a cunning trick discovered by J. Barkley Rosser in 1936 – we can drop that extra condition.

3.3 Incompleteness and incompletability

Let's concentrate for the moment on the first, semantic, version of the First Theorem.

Suppose, then, that T is a sound theory which contains the language of basic arithmetic. Then, the claim is, we can find a true G_T such that $T \nvdash \mathsf{G}_T$ and $T \nvdash \neg\mathsf{G}_T$. Let's be really clear: this doesn't, repeat *doesn't*, at all say that G_T is 'absolutely unprovable', whatever that obscure phrase could mean. It just says that G_T and its negation are *unprovable-in-T*.

Ok, you might very reasonably ask, why don't we simply 'repair the gap' in T by adding the true sentence G_T as a new axiom?

Well, consider the theory $U = T + \mathsf{G}_T$ (to use an obvious notation). Then (i) U is still sound, since the old T-axioms are true by assumption, the added new axiom is true, and the theory's logic is still truth-preserving. (ii) U is still a properly formalized theory, since adding a single specified axiom to T doesn't make it undecidable what is an axiom of the augmented theory. (iii) U's language still contains the language of basic arithmetic. So Theorem 1 still applies, and we can find a sentence G_U such that $U \nvdash \mathsf{G}_U$ and $U \nvdash \neg\mathsf{G}_U$. And since U is stronger than T we have, a fortiori, $T \nvdash \mathsf{G}_U$ and $T \nvdash \neg\mathsf{G}_U$. In other words, 'repairing the gap' in T by adding G_T as a new axiom leaves some other sentences that were undecidable in T *still* undecidable in the augmented theory.

And so it goes. Keep throwing more and more additional true axioms at T and our theory will remain negation incomplete, unless it stops being effectively axiomatized. So here's the second key take-away message of the chapter: when the conditions for Theorem 1 apply, then the theory T will not just be incomplete but in a good sense T will be *incompletable*.[4] (We'll see in due course that just the same holds when the conditions for Theorem 2 apply.)

So we should perhaps really talk of the First *Incompletability* Theorem.

3.4 The completeness and incompleteness theorems again

We have already emphasized in §2.3 the distinction we need, and we illustrated it then with a toy example. But experience suggests that it will do no harm at all to repeat the point!

Suppose T is a theory of arithmetic cast in a first-order language, and equipped with a standard first-order deductive apparatus S. Then for any φ, if T logically entails φ then $T \vdash_S \varphi$. That's Gödel's completeness theorem for S.

But T can only too easily be a negation-incomplete theory of arithmetic. Just miss out axioms for addition (say), and there can be lots of wffs φ (those involving addition) such that neither $T \vdash \varphi$ nor $T \vdash \neg\varphi$!

[4] Suppose we take a theory with *all* the true sentences of the language of basic arithmetic as axioms. Then yes, by brute force, we get a negation-complete theory! What Theorem 1 will then tell us is that this theory can't be an effectively axiomatized theory – meaning that we can't effectively decide what's an axiom, i.e. we can't effectively decide what's a true sentence of the language. We'll be soon returning to this theme.

Of course, that's a *very* boring way of being negation incomplete. And, as we said before, we might reasonably have expected that such incompleteness can always be repaired by judiciously adding in the missing axioms. What the First Incompleteness Theorem tells us, however, is that try as we might, every theory of arithmetic satisfying certain elementary and highly desirable conditions (even if it has a semantically complete logic) must *remain* negation incomplete as a theory.

4 Outlining a Gödelian proof

4.1 A notational convention

Before continuing, we should highlight a notational convention that we have already started using:

1. Expressions in informal mathematics will be in ordinary serif font, with variables, function letters etc. in *italics*. Examples:

   $$2 + 1 = 3,\ n + m = m + n,\ S(x + y) = x + Sy.$$

2. Particular expressions from formal systems – and abbreviations of them – will be in sans serif type. Examples:

   $$\mathsf{SSS0},\ \mathsf{SS0} + \mathsf{S0} = \mathsf{SSS0},\ \exists \mathsf{x}\, \mathsf{x} = \mathsf{0},\ \forall \mathsf{x} \forall \mathsf{y} (\mathsf{x} + \mathsf{y} = \mathsf{y} + \mathsf{x}).$$

3. Greek letters, like 'Σ' and 'φ', are schematic variables in the metalanguage (so, in our case, they are added to logicians' English), which we can use e.g. in generalizing about wffs of our formal systems.

In what follows, there will be a great deal of to-and-fro between (1) statements of informal mathematics, (2) formal expressions and formal proofs, and (3) general claims about formal expressions and formal proofs. It is essential for you to be clear which is which, and our (not unusual) notational convention should help you keep track.

4.2 Formally expressing numerical properties, relations and functions

In the next few sections, then, we are going to prepare the ground for §§4.6 and 4.7 where we give an outline sketch of how Gödel proved Theorem 1 (or at least, proved a very close relation).

We start with a couple more definitions. Recall, we said a language which includes the language of basic arithmetic will have (either built-in or defined) symbols '0' for zero and 'S' for the successor function. Then the standard numerals in such a language are the expressions '0', 'S0', 'SS0', 'SSS0', ...

Let's introduce a handy notational device:

Defn. 11. *We will use '$\overline{\mathsf{n}}$' to abbreviate the standard numeral denoting the natural number n.*

So '$\overline{\mathsf{n}}$' will consist of n occurrences of 'S' followed by '$\mathsf{0}$'. Hence '$\overline{\mathsf{5}}$' abbreviates '$\mathsf{SSSSS0}$' which in a formal language with standard numerals denotes what '5' denotes in informal arithmetical language.

Assume now that we are dealing with a language L which includes the language of basic arithmetic and so has standard numerals. Then we will say:

Defn. 12. *The open wff $\varphi(\mathsf{x})$ of the language L* expresses *the numerical property P just when, for any n, $\varphi(\overline{\mathsf{n}})$ is true iff n has property P.*

Similarly, the formal wff $\psi(\mathsf{x},\mathsf{y})$ expresses the numerical two-place relation R just when, for any m and n, $\psi(\overline{\mathsf{m}},\overline{\mathsf{n}})$ is true iff m has relation R to n.

And the formal wff $\chi(\mathsf{x},\mathsf{y})$ expresses the numerical one-place function f just when, for any m and n, $\chi(\overline{\mathsf{m}},\overline{\mathsf{n}})$ is true iff $f(m) = n$.

Hopefully, this definition should seem entirely natural.[1] For a couple of simple examples, the wff $\exists\mathsf{y}\,\mathsf{x} = (\mathsf{y} + \mathsf{y})$ expresses the property of being an even number. Why? Because $\exists\mathsf{y}\,\overline{\mathsf{n}} = (\mathsf{y} + \mathsf{y})$ is true just in case n is the sum of some natural number with itself, i.e. is twice some number. Similarly, $\mathsf{y} = \mathsf{x} \times \mathsf{x}$ expresses the function which squares a number, because $\overline{\mathsf{n}} = \overline{\mathsf{m}} \times \overline{\mathsf{m}}$ is true just in case $m^2 = n$.

Note, as we have defined it, for a wff to express the property of being an even number is just for it to be true of the even numbers, i.e. just for the interpreted wff to have the right *extension*. Consider the open wffs $\exists\mathsf{y}\,\mathsf{x} = ((\mathsf{S0} + \mathsf{S0}) \times \mathsf{y})$ and $\exists\mathsf{y}(\mathsf{x} = (\mathsf{y} + \mathsf{y}) \wedge \mathsf{S0} + \mathsf{S0} = \mathsf{SS0})$. These differ in intuitive sense, but again are satisfied by just the even numbers, so also count as expressing the property of being even.

The same point holds more generally: expressing a property, relation or function in our sense is just a matter of having the right extension.

The generalization of our definition to cover wffs expressing many-place relations and many-place functions is obvious: we needn't pause to spell it out.

4.3 Gödel numbers

And now for an absolutely pivotal new idea.

These days, we are entirely familiar with the fact that all kinds of data can be coded up using numbers. The idea was certainly not in such everyday currency in 1931. But even then, the following sort of definition should have looked quite unproblematic:

[1] Fine print. '$\varphi(\mathsf{x})$' indicates, of course, a wff with one or more occurrences of the variable 'x' free. But of course, the particular choice of free variable doesn't matter. '$\varphi(\overline{\mathsf{n}})$' then represents the sentence which results from replacing all free occurrences of the variable 'x' in $\varphi(\mathsf{x})$ by the standard numeral for n. As you knew!

If you've been well brought up, you might prefer the symbolism '$\varphi(\xi)$', which uses a place-holding metavariable to mark a gap, rather than use '$\varphi(\mathsf{x})$' where we are recruiting the free variable 'x' for place-holding duties. But we will stick to the more common mathematical usage (even though Fregeans will sigh sadly).

And a word to the wise: if you know what 'clash of variables' means, you will also know how we can avoid it in some future contexts by relabelling variables if necessary – so we just won't fuss about that.

Defn. 13. *A* Gödel-numbering scheme *for a formal theory T is some effective way of coding expressions of T (and sequences of expressions of T) as natural numbers. Such a scheme provides an algorithm for sending an expression (or sequence of expressions) to a number; and it also provides an algorithm for undoing the coding, sending a code number back to the unique expression (or sequence of expressions) that it codes.*

Relative to a choice of scheme, the code number for an expression (or a sequence of expressions) is its unique Gödel number*.*

For a toy example, suppose the expressions of our theory's language L are built up from just eight basic symbols. Associate those with the digits 1 to 8, and associate the comma that we might use to separate expressions in a sequence of expressions with the digit 9. Then a single L-expression, and also a sequence of L-expressions separated by commas, can be directly mapped to a sequence of digits, which can then be read as a single numeral in standard decimal notation, denoting a natural number. That mapping is the simplest of algorithms. And in reverse, undoing the coding is equally simple and mechanical – though if the string of digits expressing some number contains the digit '0', the algorithm won't output any result when we try to decode it: assume our algorithm handles such cases gracefully.

Which scheme of Gödel-numbering we adopt for theoretical purposes will be a matter of convenience. In principle nothing will hang on which we choose: any effective scheme is as good as any other (as we will be able to effectively map codes for wffs or sequences of wffs produced by one scheme to codes produced by another, simply by decoding according to the first scheme and re-coding using the second).

4.4 Three new numerical properties/relations

Defn. 14. *Take an effectively axiomatized formal theory T, and fix on a scheme for Gödel-numbering expressions and sequences of expressions from T's language. Then, relative to that numbering scheme, we can define the following properties/relations:*

> *$Wff_T(n)$ iff n is the Gödel number of a T-wff.*
> *$Sent_T(n)$ iff n is the Gödel number of a T-sentence.*
> *$Prf_T(m, n)$ iff m is the Gödel number of a T-proof of the T-sentence with code number n.*

So Wff_T, for example, is a numerical property which, so to speak, 'arithmetizes' the syntactic property of being a T-wff.

Now, these three aren't the kind of numerical properties/relations you are familiar with. But they are perfectly well-defined. Indeed, we can say more:

Theorem 3. *Suppose T is an effectively axiomatized formal theory, and suppose we are given a Gödel-numbering scheme. Then the corresponding numerical properties/relations $Wff_T, Sent_T, Prf_T$ are effectively decidable.*

Proof. [2] Consider Wff_T. The number n has this property if and only if (i) n decodes into a string of T-symbols (by an effective procedure which a computer could carry out), and (ii) that string of symbols is indeed a T-wff (which, since T has an effectively formalized language by assumption, a computer could decide). Hence it is effectively decidable whether $Wff_T(n)$.

The case of $Sent_T$ is similar. And as for Prf_T, since T is an effectively axiomatized theory it is effectively decidable whether a supposed proof-array of the theory is the genuine article proving its purported conclusion. So it is effectively decidable whether the array, if any, which gets the code number m is actually a T-proof of a sentence coded by n. That is to say, it is effectively decidable whether $Prf_T(m, n)$. ⊠

Of course, just *which* numerical relation Prf_T (for example) is will depend on the details of the theory T and on our choice of Gödel-numbering scheme. But the key point is that so long as T is an effectively axiomatized formal theory, and so long as our coding scheme is algorithm-driven too, it must be a decidable property.

4.5 T can express Prf_T

So far, so straightforward. Now things get more exciting. In this section and the next, we state two key results, which will prepare the ground for our skeleton proof of Theorem 1. For the moment, we will have to state the results without proof; later, we will see what it takes to establish them. But at this point, we just want to explain what these two key results claim.

The first is as follows:

Theorem 4. *Suppose T is an effectively axiomatized formal theory which includes the language of basic arithmetic, and suppose we have fixed on a Gödel-numbering scheme. Then T can express the corresponding numerical relation Prf_T using some arithmetical wff* $\mathsf{Prf}_T(\mathsf{x}, \mathsf{y})$.

In other words, there is a wff $\mathsf{Prf}_T(\mathsf{x}, \mathsf{y})$ in the language of basic arithmetic such that $\mathsf{Prf}_T(\overline{\mathsf{m}}, \overline{\mathsf{n}})$ is true if and only if m codes for a T-proof of the sentence with Gödel number n.

This result is *not* supposed to be obvious! So how can we prove its perhaps surprising claim?

We can take the low road. Take a particular T and trudge through the details of building a wff of basic arithmetic which indeed expresses the relation Prf_T. Then we generalize, by noting that the same strategies and tricks that we use in the chosen particular case will apply equally when dealing with other effectively axiomatized formal theories.

Or we can take the high road. We start off by showing that, quite generally, the language of basic arithmetic has the resources to express *any* decidable properties and relation. And then we apply our sweeping result to the instances we are

[2] Our end-of-proof symbol will be '⊠': we need the more usual '□' for other duties later.

interested in: for we've just seen that the numerical relation Prf_T is decidable when T is an effectively axiomatized formalized theory. We will explore a version of this option in Chapter **??**.

With a predicate Prf_T available in the theory T to express the relation Prf_T, we can now add a further simple definition:

Defn. 15. *Put* $\mathsf{Prov}_T(\mathsf{x}) =_{\text{def}} \exists \mathsf{z}\mathsf{Prf}_T(\mathsf{z}, \mathsf{x})$ *(where the quantifier, if necessary, is restricted to run over the natural numbers in the domain).*

Then $\mathsf{Prov}_T(\overline{\mathsf{n}})$*, i.e.* $\exists \mathsf{z}\mathsf{Prf}_T(\mathsf{z}, \overline{\mathsf{n}})$*, is true iff some number Gödel-numbers a* T*-proof of the sentence with Gödel-number* n*, i.e. is true just if the sentence with code number* n *is a* T*-theorem. So* $\mathsf{Prov}_T(\mathsf{x})$ *is naturally called a* provability predicate.

4.6 Defining a Gödel sentence G_T

And now comes the key result we need for building our skeletal proof of the First Theorem. Still working with an effectively axiomatized formal theory T whose language includes the language of basic arithmetic, and with a Gödel-numbering scheme in place:

Theorem 5. *We can construct a* Gödel sentence G_T *for the theory* T *in the language of basic arithmetic with the following property:* G_T *is true if and only if* $\neg\mathsf{Prov}_T(\overline{\mathsf{g}})$ *is true, where* g *is the code number of* G_T*.*

Don't worry for the moment about how we construct G_T (it is surprisingly easy). Just note at the stage what our theorem implies. By construction, we said, G_T is true on interpretation iff $\neg\mathsf{Prov}_T(\overline{\mathsf{g}})$ is true, i.e. iff the wff with Gödel number g is not a T-theorem, i.e. iff G_T is not a T-theorem. In short, our theorem tells us that we can find an arithmetical sentence G_T which is *true if and only if it isn't a* T*-theorem.*

Stretching a point, it is rather as if G_T 'says' *I am unprovable in* T. (Of course, strictly speaking, G_T doesn't *really* say that! – G_T is just a fancy sentence in the language of basic arithmetic, so it is in fact just about *numbers*, and doesn't refer to any wff. More about this later, in §**??**.) Still, stretching the point will help you to spot that we can now immediately prove ...

4.7 Incompleteness!

Here again is

Theorem 1. *Suppose* T *is an effectively axiomatized formal theory whose language contains the language of basic arithmetic. Then, if* T *is sound, there will be a true sentence* G_T *of basic arithmetic such that* $T \nvdash \mathsf{G}_T$ *and* $T \nvdash \neg\mathsf{G}_T$*, so* T *is negation incomplete.*

Proof. Take G_T to be the Gödel sentence introduced in Theorem 5. Suppose T is sound and $T \vdash \mathsf{G}_T$. Then G_T would be a theorem, and hence G_T – which is

true iff it is not a T-theorem – would be false. So T would have a false theorem and hence T would not be sound, contrary to hypothesis. So $T \nvdash \mathsf{G}_T$.

Hence G_T is not provable. Since it is true iff it is not provable, G_T is true after all. So $\neg\mathsf{G}_T$ is false and T, being sound, can't prove that either. Therefore we also have $T \nvdash \neg\mathsf{G}_T$.

So, in sum, T can't formally decide G_T one way or the other. T is negation incomplete. ⊠

This proof, once we have constructed G_T, is very straightforward. So the devil is in the details of the proofs of the preliminary results we labelled as Theorems 4 and 5. As promised, later chapters will dig down to the relevant details.

Gödel's proof of the syntactic version of the incompleteness theorem, i.e. Theorem 2, also uses the same construction of a Gödel sentence, but this time we trade in the semantic assumption that T is sound for syntactic assumptions about what T can and can't prove. So we will need syntactic analogues of Theorems 4 and 5. Again more devilish detail. Again more about this in later chapters.

4.8 Gödel and the Liar

So the claim is that, in a suitable theory T and using some Gödel coding, we can construct an arithmetic sentence G_T which as good as says that it is itself *unprovable* in T; and then such a sentence can neither be proved nor refuted in T assuming that theory is sound.

But you might well be suspicious. After all, we know we fall into paradox if we try to construct a Liar sentence L which as good as says that it is itself *not true*. So why does the construction of the Liar sentence lead to *paradox*, while the construction of the Gödel sentence gives us a *theorem*?

Which is a very good question. You have exactly the right instincts in raising it. The coming chapters, however, aim to give you a convincing answer.

But we are touching here on the deep roots of the incompleteness theorem. Suppose T is an effectively axiomatized theory which can express enough arithmetic. Then, as we'll confirm later, T can express the property of being a provable T-sentence. But, as we will also confirm, T can't express the property of being a true T-sentence (if it could, then T would be beset by the Liar paradox). So the property of being a true T-sentence and the property of being a provable T-sentence must be different properties. Hence either there are true-but-unprovable-in-T sentences or there are false-but-provable-in-T sentences. Assuming that T is sound rules out the second option. So the truths of T's language outstrip T's theorems. Therefore T can't be negation complete. *That* might be said to be the Master Argument for incompleteness: see §**??**.

5 Undecidability and incompleteness

Gödel's First Incompleteness Theorem tells us, roughly, that a nice enough theory T will always be negation incomplete for basic arithmetic.

We noted in Chapter 3 that the Theorem comes in two flavours, depending on whether we cash out the idea of being 'nice enough' in terms of (i) the semantic idea of T's being a *sound theory which uses enough of the language of arithmetic*, or (ii) the syntactic idea of T's being a *consistent theory which proves enough arithmetic*. Then we saw in Chapter 4 that Gödel's own proofs, of either flavour, go via the idea of numerically coding up syntactic facts about what can be proved in T, and then constructing an arithmetical sentence that – in virtue of the coding – is true if and only if it is not provable (it is rather as if it says *I am not provable in* T).

As we remarked, the Gödelian construction – at least as so far described – might look a bit worrying, with its echoes of the Liar Paradox. It might well go some way towards calming the worry that an illegitimate trick is being pulled if we now give a somewhat different proof of incompleteness. This proof will explicitly introduce the idea of a *diagonalization argument*. And as we will see later, it is diagonalization which is really the key to Gödel's own proof.

5.1 Negation completeness and decidability

Let's start with another definition:

Defn. 16. *A theory T is* decidable *iff the property of being a theorem of T is an effectively decidable property – i.e. iff there is a mechanical procedure for determining, for any given sentence φ of T's language, whether $T \vdash \varphi$.*

A terminology check is in order: a theory T formally *decides* a particular sentence φ iff either $T \vdash \varphi$ or $T \vdash \neg\varphi$; a theory T is *decidable* iff for *any* sentence φ of its language we can effectively determine whether $T \vdash \varphi$. Two quite different notions then, depite the similar terminology: in practice, though, you shouldn't get confused![1]

[1] To fix ideas, note that a theory can be decidable without deciding every wff. For example, the toy propositional theory T of §2.3 is decidable (as is familiar, because propositional logic is complete, a truth-table test can be used to effectively determine whether $T \vdash \varphi$ for any given wff φ of T's language). In particular, we can thereby show that $T \nvdash \mathsf{q}$ and $T \nvdash \neg\mathsf{q}$. Therefore T doesn't decide q, so T doesn't decide every wff.

Theorem 6. *Any consistent, negation-complete, effectively axiomatized formal theory is decidable.*

Proof For convenience, we can assume our theory T's proof system is a Frege/Hilbert axiomatic logic, where proofs are just linear sequences of wffs. But it should be pretty obvious how to generalize the argument to other kinds of proof systems, where proof arrays are arranged e.g. as trees of some kind.

Recall, we stipulated (in Defns. 2, 3) that if T is a properly formalized theory, its formalized language L has a finite number of basic symbols. Now, we can evidently put those basic symbols in some kind of 'alphabetical order', and then start mechanically listing off all the possible strings of symbols in order – e.g. the one-symbol strings, followed by the finite number of two-symbol strings in 'dictionary' order, followed by the finite number of three-symbol strings in 'dictionary' order, followed by the four-symbol strings, etc., etc.

Now, as we go along, generating strings of symbols, it will be a mechanical matter to decide whether a particular string is in fact a sequence of one or more wffs. And if it is, it will be a mechanical matter to decide whether the sequence of wffs is a T-proof, i.e. to check whether each wff is either an axiom or follows from earlier wffs in the sequence by one of T's rules of inference. (That's all effectively decidable in a properly formalized theory, by Defns. 2, 3). If the sequence *is* a kosher well-constructed proof, finishing with a sentence φ, then list this wff φ as a T-theorem.

We can in this way start mechanically generating a list which must eventually contain any T-theorem (since any T-theorem is the last sentence of a proof).

And that enables us to decide, of an arbitrary sentence φ of our consistent, negation-complete T, whether it is indeed a T-theorem. Just start listing all the T-theorems. Since T is negation complete, eventually either φ or $\neg\varphi$ turns up (and then you can stop!). If φ turns up, declare it to be a theorem. If $\neg\varphi$ turns up, then since T is consistent, we can declare that φ is *not* a theorem.

Hence, there *is* a dumbly mechanical 'wait and see' procedure for deciding whether φ is a T-theorem, a procedure which (given our assumptions about T) is guaranteed to deliver a verdict in a finite number of steps. ⊠

We are, of course, relying here on a *very* relaxed notion of effective decidability-in-principle, where we aren't working under any practical time constraints or constraints on available memory etc. (so note, 'effective' doesn't mean 'practically efficacious' or 'efficient'). We might have to twiddle our thumbs for an immense time before one of φ or $\neg\varphi$ turns up. Still, our 'wait and see' method is guaranteed in this case to produce a result in finite time, in an entirely mechanical way.

So this counts as an effectively computable decision procedure in our official generous sense (see again the comments on Defn. 1).

5.2 Capturing numerical properties in a theory

Here's an equivalent way of rewriting part of an earlier definition:

Defn. 12. *A numerical property P is* expressed *by the open wff $\varphi(\mathsf{x})$ with one free variable in a language L which contains the language of basic arithmetic iff, for every n,*

i. if n has the property P, then $\varphi(\overline{\mathsf{n}})$ is true,

ii. if n does not have the property P, then $\neg\varphi(\overline{\mathsf{n}})$ is true.

And now we want a new companion definition. Assume again that the language of T includes the language of basic arithmetic so can form the standard numerals. Then:

Defn. 17. *The theory T* captures *the numerical property P by the open wff $\varphi(\mathsf{x})$ iff, for any n,*

i. if n has the property P, then $T \vdash \varphi(\overline{\mathsf{n}})$,

ii. if n does not have the property P, then $T \vdash \neg\varphi(\overline{\mathsf{n}})$.

Note the contrast: what a theory can *express* depends on the richness of its language (the definition doesn't mention proofs or theorems); what a theory can *capture* – mnemonic: case-by-case prove – depends on what theorems can be derived in the theory, so depends on the richness of the theory's axioms.[2]

Just as a theory can express two-place relations (say) as well as monadic properties, a theory can capture relations as well as properties. So (for future reference) we expand our definition in the obvious way like this:

Defn 17. *(continued) The theory T captures the two-place numerical relation R by the open wff $\varphi(\mathsf{x},\mathsf{y})$ iff, for any m, n,*

i. if m has the relation R to n, then $T \vdash \varphi(\overline{\mathsf{m}},\overline{\mathsf{n}})$,

ii. if m does not have the relation R to n, then $T \vdash \neg\varphi(\overline{\mathsf{m}},\overline{\mathsf{n}})$.

But for the moment, let's concentrate on the case of capturing properties.

Ideally, of course, we will want any competent theory of arithmetic not just to express but also to capture lots of numerical properties, i.e. to be able to prove particular numbers have or lack these properties. But what kinds of properties do we want to capture?

Well, suppose that P is some effectively decidable property of numbers, i.e. one for which there is a mechanical procedure for deciding, given a natural number n, whether n has property P or not (see Defn. 1 again). So we can, in principle, run the procedure to decide whether n has this property P. Now, when we construct a formal theory of the arithmetic of the natural numbers, we will surely want deductions inside our theory to be able to track, case by case, any mechanical calculation that we can already perform informally (we see some examples of this in the next chapter). We don't want going formal to *diminish* our ability to determine whether n has the decidable numerical property P. Formalization aims at regimenting what we can in principle already do: it isn't supposed to hobble our efforts. So while we might have some passing interest in more limited

[2] To be honest, 'represents' is *much* more commonly used than my 'captures', but I'll stick here to the slightly idiosyncratic but more memorable jargon adopted in my *IGT*. Terminology here is a mess: for example, some use 'numeralwise express' to mean (not our 'express' but) 'captures/represents'.

theories, we will ideally aim for a formal theory T which at least (a) is able to frame some open wff $\varphi(\mathsf{x})$ which expresses the decidable property P, and (b) is such that if n has property P, $T \vdash \varphi(\overline{\mathsf{n}})$, and if n does not have property P, $T \vdash \neg\varphi(\overline{\mathsf{n}})$.

In short, we will want T not only to be able to *express* the decidable numerical property P but also to be able to *capture* P in the sense of our definition. Focusing on the syntactic side of this, let's say:

Defn. 18. *A formal theory T is sufficiently strong iff it captures all effectively decidable numerical properties.*[3]

Then, in summary, it seems reasonable to want a formal theory of arithmetic to be sufficiently strong. When *we* can (or at least, given world enough and time, *could*) decide of any particular number whether it has a certain property, the *theory* should be able to do that too.

5.3 Sufficiently strong theories are undecidable

We now prove a lovely theorem (take the elegant proof slowly, savour it!):

Theorem 7. *No consistent, effectively axiomatized and sufficiently strong formal theory is decidable.*

Proof We suppose T is a consistent and sufficiently strong theory yet also decidable, and derive a contradiction.

If T is sufficiently strong, it must have a supply of open wffs suitable for capturing numerical properties. And in fact, by Defn 2, it must be decidable what strings of symbols are T-wffs with the free variable 'x'. So, we can use the same idea as in the proof of Theorem 6 to start mechanically listing such wffs

$$\varphi_0(\mathsf{x}), \varphi_1(\mathsf{x}), \varphi_2(\mathsf{x}), \varphi_3(\mathsf{x}), \ldots.$$

For we can just start churning out all the strings of symbols of T's language (by length and in 'alphabetical order'), and as we go along we mechanically select out the wffs with free variable 'x'.

We can then introduce the following definition of the numerical property D:

(*) n has the property D if and only if $T \vdash \neg\varphi_n(\overline{\mathsf{n}})$.

That's a perfectly coherent stipulation. Of course, property D isn't presented in the familiar way in which we ordinarily present properties of numbers: but our definition tells us what has to be the case for n to have the property D, and that's all we will need.

Now for the key observation: our supposition that T is a decidable theory entails that D is an effectively decidable property of numbers.

Why? Well, given any number n, it will be a mechanical matter to start listing off the open wffs until we get to the n-th one, $\varphi_n(\mathsf{x})$. Then it is a mechanical

[3] It would be equally natural, of course, to require that the theory also capture all decidable relations and all computable functions – but for present purposes we don't need to add that.

matter to form the numeral $\overline{\mathsf{n}}$, substitute it for the variable, and then prefix a negation sign. Now we just apply the supposed mechanical procedure for deciding whether a sentence is a T-theorem to test whether the resulting wff $\neg\varphi_n(\overline{\mathsf{n}})$ is a theorem. So, on our current assumptions, there is an algorithm for deciding whether n has the property D.

Since, by hypothesis, the theory T is sufficiently strong, it can capture all decidable numerical properties. Hence it follows, in particular, that D is capturable by some open wff. This wff must of course eventually occur somewhere in our list of the $\varphi(\mathsf{x})$. Let's suppose the d-th wff does the trick: that is to say, property D is captured by $\varphi_d(\mathsf{x})$.

It is now entirely routine to get out a contradiction. For, just by the definition of capturing, to say that $\varphi_d(\mathsf{x})$ captures D means that for any n,

> if n has the property D, $T \vdash \varphi_d(\overline{\mathsf{n}})$,
> if n doesn't have the property D, $T \vdash \neg\varphi_d(\overline{\mathsf{n}})$.

So taking in particular the case $n = d$, we have

i. if d has the property D, $T \vdash \varphi_d(\overline{\mathsf{d}})$,
ii. if d doesn't have the property D, $T \vdash \neg\varphi_d(\overline{\mathsf{d}})$.

But note what our initial definition (*) of the property D implies for the particular case $n = d$:

iii. d has the property D if and only if $T \vdash \neg\varphi_d(\overline{\mathsf{d}})$.

From (ii) and (iii), it follows that whether d has property D or not, the wff $\neg\varphi_d(\overline{\mathsf{d}})$ is a theorem either way. So by (iii) again, d does have property D, hence by (i) the wff $\varphi_d(\overline{\mathsf{d}})$ must be a theorem too. So a wff and its negation are both theorems of T. Which makes T inconsistent.

In sum, the supposition that T is a consistent and sufficiently strong axiomatized formal theory *and* is decidable leads to contradiction. ⊠

5.4 A word about 'diagonalization'

Let's highlight the key construction here. In defining the property D, for each n, we take the n-th wff $\varphi_n(\mathsf{x})$ in our list, and plug in the standard numeral for the place-index n (before taking the negation of the result). This sort of thing is called *diagonalization*. Why? Just consider the square array you get by writing

$$\begin{array}{cccccc}
\varphi_0(\overline{\mathsf{0}}) & \varphi_0(\overline{\mathsf{1}}) & \varphi_0(\overline{\mathsf{2}}) & \varphi_0(\overline{\mathsf{3}}) & \dots \\
\varphi_1(\overline{\mathsf{0}}) & \varphi_1(\overline{\mathsf{1}}) & \varphi_1(\overline{\mathsf{2}}) & \varphi_1(\overline{\mathsf{3}}) & \dots \\
\varphi_2(\overline{\mathsf{0}}) & \varphi_2(\overline{\mathsf{1}}) & \varphi_2(\overline{\mathsf{2}}) & \varphi_2(\overline{\mathsf{3}}) & \dots \\
\varphi_3(\overline{\mathsf{0}}) & \varphi_3(\overline{\mathsf{1}}) & \varphi_3(\overline{\mathsf{2}}) & \varphi_3(\overline{\mathsf{3}}) & \dots \\
\dots & \dots & \dots & \dots & \searrow
\end{array}$$

Evidently, the wffs of the form $\varphi_n(\overline{n})$, including $\varphi_d(\overline{d})$, lie down the diagonal through the array.

We'll be meeting other instances of this sort of diagonal construction. And it is a diagonalization of this kind that is really at the heart of Gödel's incompleteness proof.[4] More about this in due course.

5.5 Incompleteness again!

So we have now shown:

Theorem 6. *Any consistent, negation-complete, effectively axiomatized formal theory is decidable.*

Theorem 7. *No consistent, effectively axiomatized and sufficiently strong formal theory is decidable.*

We can therefore immediately deduce:

Theorem 8. *A consistent, effectively axiomatized, sufficiently strong, formal theory cannot be negation complete.*

Wonderful! A seemingly remarkable theorem, proved remarkably quickly (this time without having to simply assume some as-yet-unproved theorems along the way).[5]

Note, though, that – unlike Gödel's own proof strategy – Theorem 8 doesn't actually yield a specific undecidable sentence for a given theory T. And more importantly, the interest of the theorem depends on the still-informal notion of a 'sufficiently strong' theory being in good order. Have we perhaps just shown that looking for sufficient strength is, after all, an unreasonable demand?

Now, I wouldn't have written up the argument in this chapter if this notion of T's being sufficiently strong were intrinsically problematic. Still, we are left with a major task here: we will need to give a sharper account of what makes for an effectively decidable property in order to (i) clarify the notion of sufficient strength, while (ii) making it plausible that we really do want theories to be sufficiently strong in this clarified sense.

This can be done. However, supplying and defending the needed sharp account of the notion of effective decidability takes quite a bit of work. And we don't need to do the work in order to prove core versions of the First Incompleteness Theorem via Gödel's original method as partially sketched in Chapter 4. So, over the coming chapters, we are going to start by reverting to exploring something closer to Gödel's route to the incompleteness theorems.

[4] The grandfather of all such uses of diagonalization is Cantor's diagonal argument to show a set can't be equinumerous with its powerset (see e.g. the Wikipedia entry, as well as *IGT2*, §2.5).

[5] I learnt the argument for Theorem 8 as a student – so decades ago! – from lectures by Timothy Smiley.