1 What Gödel's Theorems say

1.1 Basic arithmetic

It seems to be child's play to grasp the fundamental notions involved in the arithmetic of addition and multiplication. Starting from zero, there is a sequence of 'counting' numbers, each having just one immediate successor. This sequence of numbers – officially, *the natural numbers* – continues without end, never circling back on itself; and there are no 'stray' numbers, lurking outside this sequence. Adding n to m is the operation of starting from m in the number sequence and moving n places along. Multiplying m by n is the operation of (starting from zero and) repeatedly adding m, n times. And that's about it.

Once these fundamental notions are in place, we can readily define many more arithmetical notions in terms of them. Thus, for any natural numbers m and n, $m < n$ iff there is a number $k \neq 0$ such that $m + k = n$. m is a factor of n iff $0 < m$ and there is some number k such that $0 < k$ and $m \times k = n$. m is even iff it has 2 as a factor. m is prime iff $1 < m$ and m's only factors are 1 and itself. And so on.[1]

Using our basic and/or defined concepts, we can then make various general claims about the arithmetic of addition and multiplication. There are familiar truths like 'addition is commutative', i.e. for any numbers m and n, we have $m + n = n + m$. There are also yet-to-be-proved conjectures like Goldbach's conjecture that every even number greater than two is the sum of two primes.

That second example illustrates the truism that it is one thing to understand what we'll call *the language of basic arithmetic* (i.e. the language of the addition and multiplication of natural numbers, together with the standard first-order logical apparatus), and it is another thing to be able to evaluate claims that can be framed in that language.

Still, it is extremely plausible to suppose that, whether the answers are readily available to us or not, questions posed in the language of basic arithmetic do *have* entirely determinate answers. The structure of the number sequence is (surely) simple and clear. There's a single, never-ending sequence, starting with zero; each number is followed by a unique successor; each number is reached by a finite number of steps from zero; there are no repetitions. The operations of addition and multiplication are again (surely) entirely determinate; their outcomes are fixed by the school-room rules. So what more could be needed to fix the truth or falsity of propositions that – perhaps via a chain of definitions – amount to claims of basic arithmetic? To put it fancifully: God sets down the number sequence

[1] 'Iff' is, of course, the standard logicians' shorthand for 'if and only if'.

and specifies how the operations of addition and multiplication work. He has then done all he needs to do to make it the case that Goldbach's conjecture is true (or false, as the case may be).

Of course, that last remark is *far* too fanciful for comfort. We may find it compelling to think that the sequence of natural numbers has a definite structure, and that the operations of addition and multiplication are entirely nailed down by the familiar school-room rules. But what is the real content of the thought that the truth-values of all basic arithmetic propositions are thereby 'fixed'?

Here's one initially attractive way of giving non-metaphorical content to that thought. The idea is that we can specify a bundle of fundamental assumptions or *axioms* which somehow pin down the structure of the number sequence, and which also characterize addition and multiplication (after all, it is entirely natural to suppose that we *can* give a reasonably simple list of true axioms to encapsulate the fundamental principles so readily grasped by the successful learner of school arithmetic). So suppose that φ is a proposition which can be formulated in the language of basic arithmetic. Then, the plausible suggestion continues, the assumed truth of our axioms always 'fixes' the truth-value of any such φ in the following sense: either φ is logically deducible from the axioms by a normal kind of proof, and so φ is true; or its negation $\neg\varphi$ is deducible from the axioms, and so φ is false.[2] We may not, of course, actually stumble on a proof one way or the other: but the idea is that such a proof always exists, since the axioms contain enough information to enable the truth-value of any basic arithmetical proposition to be deductively extracted by deploying familiar step-by-step logical rules of inference.

Logicians say that a theory T is *(negation) complete* if, for every sentence φ in the language of the theory, either φ or $\neg\varphi$ is deducible in T's proof system. So, put into that jargon, the suggestion we are considering is this: we should be able to specify a reasonably simple bundle of true axioms which, together with some logic, give us a *complete* theory of basic arithmetic: we could in principle use the theory to prove the truth or falsity of any claim about addition and/or multiplication (or at least, any claim we can state using quantifiers like 'for all', connectives like 'if' and 'not', and identity). And if that's right, truth in basic arithmetic could just be equated with provability in this complete theory.

It is tempting to say more. For what will the axioms of basic arithmetic look like? Here's one candidate: 'For every natural number, there's a unique next one'. This is evidently true: but evident *how*? Is it that we have some special and rather mysterious faculty of mathematical intuition which allows us just to *'see'* that this axiom is true? Or can we avoid an appeal to intuition? Maybe the axiom is evidently true because it is some kind of definitional triviality. Perhaps it is just part of what we *mean* by talk of the natural numbers that we are dealing with an ordered sequence where each member of the sequence has a

[2] 'Normal proof' is vague, and soon we will need to be more careful: but the idea is that we don't want to countenance, e.g., 'proofs' with an infinite number of steps.

unique successor. And, plausibly, other candidate axioms are similarly true by definition (or are logically derivable from definitions).

If those tempting thoughts are right – if the truths of basic arithmetic all flow deductively from logic plus definitionally true axioms – then true arithmetical claims would be simply *analytic* in the philosophers' sense.[3] And this so-called 'logicist' view would then give us a very neat explanation of the special certainty and the necessary truth of correct claims of basic arithmetic.

1.2 Incompleteness

But now, in headline terms, *Gödel's First Incompleteness Theorem shows that the entirely natural idea that we can completely axiomatize basic arithmetic is wrong.* Suppose we try to specify a suitable axiomatic theory T that seems to capture the structure of the natural number sequence and pin down addition and multiplication (and maybe a lot more besides). Then Gödel gives us a recipe for coming up with a corresponding sentence G_T, couched in the language of basic arithmetic, such that (i) we can show (on very modest assumptions, e.g. that T is consistent) that neither G_T nor $\neg\mathsf{G}_T$ can be derived in T, and yet (ii) we can also recognize that, at least if T *is* consistent, G_T will be true.

This is surely astonishing. Somehow, it seems, the class of basic arithmetic truths about addition and multiplication will *always* elude our attempts to pin it down by a fixed set of fundamental assumptions from which we can deduce everything else.

How does Gödel show this in his great 1931 paper which presents the Incompleteness Theorems? Well, note how we can use numbers and numerical propositions to encode facts about all sorts of things. For a trivial example, students in the philosophy department might be numbered off in such a way that one student's code-number is less than another's if the first student enrolled before than the second; a student's code-number ends with '1' if she is an undergraduate student and with '2' if she is a graduate; and so on and so forth. More excitingly, we can use numbers and numerical propositions to encode facts about theories, e.g. facts about what can be derived in a theory T.[4] And

[3]Thus Gottlob Frege, writing in his wonderful *Grundlagen der Arithmetik*, urges us to seek the proof of a mathematical proposition by 'following it up right back to the primitive truths. If, in carrying out this process, we come only on general logical laws and on definitions, then the truth is an analytic one.' (Frege, 1884, p. 4)

[4]It is absolutely standard for logicians to talk of a theory T as *proving* a sentence φ when there is a logically correct derivation of φ from T's assumptions. But T's assumptions may be contentious or plain false or downright absurd. So, T's proving φ in the logician's sense does not mean that φ is proved in the sense that it is established as true. It is far too late in the game to kick against the logician's usage, and in most contexts it is harmless. But our special concern in this book is with the connections and contrasts between being true and being provable in this or that theory T. So we need to be on our guard. And to help emphasize that proving-in-T is not always proving-as-true, I'll often talk of 'deriving' rather than 'proving' sentences when it is the logician's notion which is in play.

what Gödel did is find a general method that enabled him to take any theory T strong enough to capture a modest amount of basic arithmetic and construct a corresponding arithmetical sentence G_T which encodes the claim 'The sentence G_T itself is unprovable in theory T'. So G_T is true if and only if T can't prove it.

Suppose that T has true axioms and a reliably truth-preserving deductive logic. Then everything T proves must be true, i.e. T is a *sound* theory. But if T were to prove its Gödel sentence G_T, then it would prove a falsehood (since G_T is true if and only if it is unprovable). Hence, if T is sound, G_T is unprovable in T. But then G_T is *true*. Hence $\neg\mathsf{G}_T$ is false; and so that too can't be proved by T, because T only proves truths. In sum, still assuming T is sound, neither G_T nor its negation will be provable in T: therefore T can't be negation complete. And in fact we don't even need to assume that T is sound: the official First Theorem shows, for a start, that T's mere consistency is enough to guarantee that a suitably constructed G_T is true-but-unprovable-in-T.

To repeat: the sentence G_T encodes the claim that that very sentence is unprovable. But doesn't this make G_T uncomfortably reminiscent of the Liar sentence 'This very sentence is false' (which is false if it is true, and true if it is false)? You might well wonder whether Gödel's argument doesn't lead to a cousin of the Liar paradox rather than to a theorem. But not so. As we will soon see, there is nothing at all suspect or paradoxical about Gödel's First Theorem as a technical result about formal axiomatized systems (a result which we can in any case prove without appeal to 'self-referential' sentences).

'Hold on! If we can locate G_T, a Gödel sentence for our favourite nicely axiomatized theory of arithmetic T, and can argue that G_T is true-but-unprovable, why can't we just patch things up by adding it to T as a new axiom?' Well, to be sure, if we start off with theory T (from which we can't deduce G_T), and add G_T as a new axiom, we'll get an expanded theory $U = T + \mathsf{G}_T$ from which we *can* quite trivially derive G_T. But we can now just re-apply Gödel's method to our improved theory U to find a new true-but-unprovable-in-U arithmetic sentence G_U that encodes 'I am unprovable in U'. So U again is incomplete. Thus T is not only incomplete but, in a quite crucial sense, is *incompletable*.

Let's emphasize this key point. There's nothing mysterious about a theory failing to be negation complete, plain and simple. Imagine the departmental administrator's 'theory' D which records some basic facts about the course selections of a group of students: the language of D, let's suppose, is very limited and can only be used to tell us about who takes what course in what room when. From the 'axioms' of D we'll be able, let's suppose, to deduce further facts – such as that Jack and Jill take a course together, and that ten people are taking the logic course. But if there's no relevant axiom in D about their classmate Jo, we might not be able to deduce either J = 'Jo takes logic' or $\neg\mathsf{J}$ = 'Jo doesn't take logic'. In that case, D isn't yet a negation-complete story about the course selections of students. However, that's just boring: for the 'theory' about course selection is no doubt completable (i.e. it can be expanded to settle every question that can be posed in its very limited language). By contrast, what gives Gödel's First Theorem its real bite is that it shows that any properly axiomatized and consistent theory of basic arithmetic must *remain* incomplete, whatever our efforts to complete it by throwing further axioms into the mix.

Finally, note that since G_U can't be derived from U, i.e. $T + \mathsf{G}_T$, it can't be derived from the original T either. So we can iterate the same Gödelian construction to generate a never-ending stream of independent true-but-unprovable sentences for any nicely axiomatized T including enough basic arithmetic.

1.3 More incompleteness

Incompletability does not just affect theories of basic arithmetic. Consider set theory, for example. Start with the empty set $\varnothing$. Form the set $\{\varnothing\}$ containing $\varnothing$ as its sole member. Now form the set $\{\varnothing, \{\varnothing\}\}$ containing the empty set we started off with plus the set we've just constructed. Keep on going, at each stage forming the set of all the sets so far constructed. We get the sequence

$$\varnothing, \{\varnothing\}, \{\varnothing, \{\varnothing\}\}, \{\varnothing, \{\varnothing\}, \{\varnothing, \{\varnothing\}\}\}, \ldots$$

This sequence has the structure of the natural numbers. We can pick out a first member (corresponding to zero); each member has one and only one successor; it never repeats. We can go on to define analogues of addition and multiplication. Moreover, any standard set theory can define this sequence. So if we could have a negation-complete axiomatized set theory, then we could, in particular, have a negation-complete theory of the fragment of set theory which provides us with an analogue of arithmetic; adding a simple routine for translating the results for this fragment into the familiar language of basic arithmetic would then give us a complete theory of arithmetic. Hence, by Gödel's First Incompleteness Theorem, there cannot be a negation-complete set theory.

The point evidently generalizes: any axiomatized mathematical theory T that can define (an analogue of) the natural-number sequence and replicate enough of the basic arithmetic of addition and multiplication must be incomplete and incompletable.[5]

1.4 Some implications?

Gödelian incompleteness immediately defeats what is otherwise a surely attractive suggestion about the status of arithmetic – namely the logicist idea that it all flows deductively from a simple bunch of definitional truths that articulate the very ideas of the natural numbers, addition and multiplication.

But then, how *do* we manage somehow to latch on to the nature of the unending number sequence and the operations of addition and multiplication in a way that outstrips whatever rules and principles can be captured in definitions?

[5]We return to this point more carefully in Section 18.2.

At this point it can seem that we must have a rule-transcending cognitive grasp of the numbers which underlies our ability to recognize certain 'Gödel sentences' as correct arithmetical propositions. And if you are tempted to think so, then you may well be further tempted to conclude that minds such as ours, capable of such rule-transcendence, can't be machines (supposing, reasonably enough, that the cognitive operations of anything properly called a machine can be fully captured by rules governing the machine's behaviour).

So there's apparently a quick route from reflections about Gödel's First Theorem to some conclusions about the nature of arithmetical truth and the nature of the minds that grasp it. Whether those conclusions really follow will emerge later. For the moment, we have an initial idea of what the Theorem says and why it might matter – enough, I hope, already to entice you to delve further into the story that unfolds in this book.

1.5 The unprovability of consistency

If we can derive even a modest amount of basic arithmetic in theory T, then we'll be able to derive $0 \neq 1$.[6] So if T *also* proves $0 = 1$, it is inconsistent. Conversely, if T is inconsistent, then – since we can derive anything in an inconsistent theory[7] – it can prove $0 = 1$. But we said that we can use numerical propositions to encode facts about what can be derived in T. So there will in particular be a *numerical* consistency sentence Con_T that encodes the claim that we can't derive $0 = 1$ in T, i.e. encodes in a natural way the claim that T is consistent.

We know, however, that there is a numerical proposition which encodes the claim that G_T is unprovable: we have already said that it is G_T itself.

So this means that (part of) the conclusion of Gödel's First Theorem, namely the claim that if T is consistent, then G_T is unprovable, can *itself* be encoded by a numerical proposition, namely $\mathsf{Con}_T \rightarrow \mathsf{G}_T$. And now for another wonderful Gödelian insight. It turns out that the informal reasoning that we use, outside T, to show 'if T is consistent, then G_T is unprovable' is elementary enough to be mirrored by reasoning inside T (i.e. by reasoning with numerical propositions which encode facts about T-proofs). Or at least that's true so long as T satisfies conditions only slightly stronger than the First Theorem assumes. So, again on modest assumptions, we can derive $\mathsf{Con}_T \rightarrow \mathsf{G}_T$ inside T.

But the First Theorem has already shown that if T is consistent we can't derive G_T in T. So it immediately follows that if T is consistent it can't prove Con_T. And *that* is Gödel's Second Incompleteness Theorem. Roughly interpreted: nice theories that include enough basic arithmetic can't prove their own consistency.[8]

[6] We'll allow ourselves to abbreviate expressions of the form $\neg\sigma = \tau$ as $\sigma \neq \tau$.

[7] There are, to be sure, deviant non-classical logics in which this principle doesn't hold. In this book, however, we aren't going to take further note of them, if only because of considerations of space.

[8] That *is* rough. The Second Theorem shows that T can't prove Con_T, which is certainly *one* natural way of expressing T's consistency inside T. But couldn't there perhaps be some

1.6 More implications?

Suppose that there's a genuine issue about whether T is consistent. Then even before we'd ever heard of Gödel's Second Theorem, we wouldn't have been convinced of its consistency by a derivation of Con_T inside T. For we'd just note that if T were in fact inconsistent, we'd be able to derive any T-sentence we like in the theory – including a statement of its own consistency!

The Second Theorem now shows that we would indeed be right not to trust a theory's announcement of its own consistency. For (assuming T includes enough arithmetic), if T entails Con_T, then the theory must in fact be *in*consistent.

However, the real impact of the Second Theorem isn't in the limitations it places on a theory's proving its own consistency. The key point is this. If a nice arithmetical theory T can't even prove *itself* to be consistent, it certainly can't prove that a *richer* theory T^+ is consistent (since if the richer theory is consistent, then any cut-down part of it is consistent). Hence we can't use 'safe' reasoning of the kind we can encode in ordinary arithmetic to prove other more 'risky' mathematical theories are in good shape. For example, we can't use unproblematic arithmetical reasoning to convince ourselves of the consistency of set theory (with its postulation of a universe of wildly infinite sets).

And *that* is a very interesting result, for it seems to sabotage what is called Hilbert's Programme, which is precisely the project of defending the wilder reaches of infinitistic mathematics by giving consistency proofs which use only 'safe' methods. A lot more about this in due course.

1.7 What's next?

What we've said so far, of course, has been very sketchy and introductory. We must now start to do better. In Chapter 2, we introduce the notions of effective computability, decidability and enumerability, notions we are going to need in what follows. Then in Chapter 3, we explain more carefully what we mean by talking about an 'axiomatized theory' and prove some elementary results about axiomatized theories in general. In Chapter 4, we introduce some concepts relating specifically to axiomatized theories of arithmetic. Then in Chapters 5 and 6 we prove a pair of neat and relatively easy results – namely that any sound and 'sufficiently expressive' axiomatized theory of arithmetic, and likewise any consistent and 'sufficiently strong' axiomatized theory, is negation incomplete. For reasons that we'll explain, these informal results fall some way short of Gödel's own First Incompleteness Theorem. But they do provide a very nice introduction to some key ideas that we'll be developing more formally in the ensuing chapters.

other sentence of T, Con'_T, which also in some good sense expresses T's consistency, where T doesn't prove $\mathsf{Con}'_T \to \mathsf{G}_T$ but *does* prove Con'_T? We'll return to this question in Sections 24.5 and 27.2.

2 Decidability and enumerability

This chapter briskly introduces a number of concepts – mostly related to the idea of computability – that we'll need in the next few chapters. Later in the book, we'll return to some of these ideas and give sharper, technical, treatments of them. But for present purposes, informal intuitive presentations are enough.

2.1 Functions

We'd better start, however, by very quickly reviewing some standard jargon and notation for talking about functions, since functions will feature so prominently in what follows. For simplicity, we'll focus here on one-place functions (it will be obvious how to generalize definitions to cover many-place functions).

Our concern will be with *total* functions $f\colon \Delta \to \Gamma$, i.e. with functions which map *every* element x of the *domain* Δ to exactly one corresponding value $f(x)$ in the set Γ.[1] We then say

i. The *range* of a function $f\colon \Delta \to \Gamma$ is $\{f(x) \mid x \in \Delta\}$, i.e. the set of elements in Γ that are values of f for arguments in Δ.

ii. A function $f\colon \Delta \to \Gamma$ is *surjective* iff the range of f is the whole of Γ – i.e. if for every $y \in \Gamma$ there is some $x \in \Delta$ such that $f(x) = y$. (If you prefer that in English, you can say that such a function is *onto*, since it maps Δ onto the whole of Γ.)

iii. A function $f\colon \Delta \to \Gamma$ is *injective* iff f maps different elements of Δ to different elements of Γ – i.e. if $x \neq y$ then $f(x) \neq f(y)$. (If you prefer that in English, you can say that such a function is *one-to-one*.)

iv. A function $f\colon \Delta \to \Gamma$ is *bijective* if it is both surjective and injective. (In English again, f is then a *one-one correspondence* between Δ and Γ.)

2.2 Effective decidability, effective computability

(a) Familiar school-room arithmetic routines (e.g. for testing whether a number is prime) give us ways of *effectively deciding* whether some property holds. Other

[1] For wider mathematical purposes, the more general idea of a *partial* function becomes essential. This is a mapping f which is not necessarily defined for all elements of its domain (for an obvious example, consider the reciprocal function $1/x$ for rational numbers, which is not defined for $x = 0$). However, we won't need to say much about partial functions in this book, and hence – by default – plain 'function' will henceforth always mean 'total function'.

routines (e.g. for squaring a number or finding the highest common factor of two numbers) give us ways of *effectively computing* the value of a function.

What is meant by talking of *effective* procedures? Well, we are trying to sharpen the otherwise rather vague, intuitive, notion of a computation. And the core idea is that an effective procedure involves executing an *algorithm* which *successfully terminates*.

Here, an algorithm is a set of step-by-step instructions (instructions which are pinned down in advance of their execution), with each small step clearly specified in every detail (leaving no room for doubt as to what does and what doesn't count as executing the step). More carefully, executing an algorithm (i) involves an entirely determinate sequence of discrete step-by-small-step procedures (where each small step is readily executable by a very limited calculating agent or machine). (ii) There isn't any room left for the exercise of imagination or intuition or fallible human judgement. Further, in order to execute the procedures, (iii) we don't have to resort to outside 'oracles' (i.e. independent sources of information), and (iv) we don't have to resort to random methods (coin tosses). Such algorithmic procedures can be followed by a dumb computer. Indeed, it is natural to turn this observation into a first shot at an informal definition:

> An algorithmic procedure is one that a suitably programmed computer can execute.

But plainly, if an algorithmic procedure is actually to decide whether some property holds or actually to compute a function, more is required. It needs to terminate after a finite number of steps and deliver a result!

So, putting these ideas together, we can give two interrelated rough definitions:

> A property/relation is *effectively decidable* iff there is an algorithmic procedure that a suitably programmed computer could use to decide, in a finite number of steps, whether the property/relation applies in any given case.
>
> A total function is *effectively computable* iff there is an algorithmic procedure that a suitably programmed computer could use for calculating, in a finite number of steps, the value of the function for any given argument.[2]

(b) But what kind of computer do we have in mind here when we gesture towards a definition by saying that an algorithmic procedure is one that a computer can execute? We need to say something more about the relevant sort of computer's *size and speed*, and *architecture*.

A real-life computer is limited in size and speed. There will be some upper bound on the size of the inputs it can handle; there will be an upper bound on the size of the set of instructions it can store; there will be an upper bound on

[2]For more about how to relate these two definitions via the notion of a 'characteristic function', see Section 11.6.

the size of its working memory. And even if we feed in inputs and instructions it can handle, it is of little practical use to us if the computer won't finish executing its algorithmic procedure for centuries.

Still, we are cheerfully going to abstract from all these 'merely practical' considerations of size and speed – which is why we said nothing about them in explaining what we mean by effective procedures. In other words, we will count a function as being effectively computable if there is a finite set of step-by-step instructions which a computer could *in principle* use to calculate the function's value for any particular arguments, given memory, working space and time enough. Likewise, we will say that a property is effectively decidable if there is a finite set of step-by-step instructions a computer can use which is in principle guaranteed to decide whether the property applies in any given case, again abstracting from worries about limitations of time and memory. Let's be clear, then: 'effective' here does *not* mean that the computation must be feasible for us, on existing computers, in real time. So, for example, we count a numerical property as effectively decidable in this broad sense even if on existing computers it might take longer to compute whether a given number has it than we have time left before the heat death of the universe. It is enough that there's an algorithm that works in theory and would deliver an answer in the end, if only we had the computational resources to use it and could wait long enough.

'But then,' you might well ask, 'why on earth bother with these radically idealized notions of computability and decidability? If we allow procedures that may not deliver a verdict in the lifetime of the universe, what good is that? If we are interested in issues of computability, shouldn't we really be concerned not with idealized-computability-in-principle but with some stronger notion of *practicable* computability?'

That's a fair challenge. And modern computer science has much to say about grades of computational complexity and levels of feasibility. However, we will stick to our ultra-idealized notions of computability and decidability. Why? Because later we'll be proving a range of limitative theorems, e.g. about what can't be algorithmically decided. By working with a very weak 'in principle' notion of what is required for being decidable, our impossibility results will be correspondingly very strong – they won't depend on any mere contingencies about what is practicable, given the current state of our software and hardware, and given real-world limitations of time or resources. They show that some problems can't be mechanically decided, even on the most generous understanding of that idea.

(c) We've said that we are going to be abstracting from limitations on storage, etc. But you might suspect that this still leaves much to be settled. Doesn't the 'architecture' of a computing device affect what it can compute?

The short answer is that it doesn't (at least, once we are dealing with devices of a certain degree of complexity, which can act as 'general purpose' computers). And intriguingly, some of the central theoretical questions here were the subject