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

Preface		vii
1	A very brief note on Kurt Gödel	1
2	Incompleteness, the very idea	2
3	The First Theorem, two versions	11
4	Outlining a Gödelian proof	15
5	Undecidability and incompleteness	21
6	Two weak arithmetics	27
7	First-order Peano Arithmetic	40
8	Quantifier complexity	48
Interlude		52

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.*

'$\Sigma \vDash \varphi$' 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: $(\mathsf{p} \land \neg\mathsf{r})$.

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?

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.

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 '$\mathsf{0}$' for zero and 'S' for the successor function. Then the standard numerals in such a language are the expressions '$\mathsf{0}$', '$\mathsf{S0}$', '$\mathsf{SS0}$', '$\mathsf{SSS0}$', ...

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 very well 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 familiar 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:*

> $\mathit{Wff}_T(n)$ *iff n is the Gödel number of a T-wff.*
> $\mathit{Sent}_T(n)$ *iff n is the Gödel number of a T-sentence.*
> $\mathit{Prf}_T(m, n)$ *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 $\mathit{Wff}_T, \mathit{Sent}_T, \mathit{Prf}_T$ 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 $\mathit{Wff}_T(n)$.

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 $\mathit{Prf}_T(m, n)$. ⊠

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

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

Evidently, the wffs of the form $\varphi_n(\overline{\mathsf{n}})$, including $\varphi_d(\overline{\mathsf{d}})$, 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.

6 Two weak arithmetics

So far we have talked rather abstractly of theories which 'can prove a certain modest amount of arithmetic' and about theories which are 'sufficiently strong'. But we haven't said anything about what such theories look like. It is obviously high time that we stopped operating at the level of abstraction of earlier chapters; we need to start getting down to details.

This chapter, then, introduces a couple of weak arithmetics ('arithmetics', that is to say, in the sense of 'theories of arithmetic'). We first meet Baby Arithmetic and then the important Robinson Arithmetic. You can by all means skip lightly over a few of the more boring proof details here; but you do need to get a sense of how these two weaker formal theories work, in preparation for the next chapter where we introduce the much stronger Peano Arithmetic.

6.1 The language L_B

First we describe *the language of baby arithmetic*, L_B. Its symbols, with their built-in interpretations, are

$\mathsf{0}$	constant, denoting zero
$\mathsf{S}, +, \times$	function symbols for, respectively, the successor, addition and multiplication functions
$=, \neg, \to$	the identity predicate, negation, and conditional
$(,)$	parentheses for use with $+$, $\times$ and $\to$.

We could give our language the other propositional connectives if we like: but the crucial thing is that L_B lacks the apparatus of quantification.

We write the one-place successor function symbol in prefix position, so we can form the standard numerals $\mathsf{0}, \mathsf{S0}, \mathsf{SS0}, \mathsf{SSS0}, \ldots$ (see §2.5). Recall, we use '$\overline{\mathsf{n}}$' to represent the standard numeral $\mathsf{SS} \ldots \mathsf{S0}$ with n occurrences of 'S'.

We will however write '$+$' and '$\times$' as infix function symbols in the usual way – i.e. we write $(\mathsf{S0} + \mathsf{SS0})$ rather than prefix the function sign as in $+\mathsf{S0SS0}$. So we need the parentheses for scoping the function signs, to disambiguate $\mathsf{S0} + \mathsf{SS0} \times \mathsf{SSS0}$, e.g. as $(\mathsf{S0} + (\mathsf{SS0} \times \mathsf{SSS0}))$. For readability, though, we will follow common practice and usually drop outermost pairs of brackets.

From these symbols, we can construct the *terms* of L_B. A term is a referring expression built up from occurrences of '$\mathsf{0}$' and applications of the function expressions 'S', '+', '×'. Examples are $\mathsf{0}$, $\mathsf{SSS0}$, $\mathsf{S0} + \mathsf{SS0}$, $(\mathsf{S0} + \mathsf{SS0}) \times \mathsf{SSS0}$, $\mathsf{SSS0} + ((\mathsf{S0} + \mathsf{SS0}) \times \mathsf{SSS0})$, and so on.

We will use σ and τ throughout this chapter as metalinguistic placeholders for terms. The *value* of a term τ is the number it denotes when standardly interpreted: the values of our example terms are respectively 0, 3, 3, 9 and 12.

The sole built-in predicate of the language L_B is the identity sign. Since L_B lacks non-logical predicates, the only way of forming atomic wffs in the language is therefore by taking two terms and putting the identity sign between them. In other words, the atomic wffs of L_B are *equations* relating terms denoting particular numbers. So, for example, $\mathsf{S0} + \mathsf{SS0} = \mathsf{SSS0}$ is a true atomic wff – which we can abbreviate as $\overline{1} + \overline{2} = \overline{3}$. And $\mathsf{S0} + \mathsf{SS0} = \mathsf{SS0} \times \mathsf{SS0}$ is a false atomic wff – which we can abbreviate as $\overline{1} + \overline{2} = \overline{2} \times \overline{2}$.

We now add a negation sign to the language L_B so that we can also explicitly assert that various equations do *not* hold. For example, $\neg\, \mathsf{S0} + \mathsf{SS0} = \mathsf{SS0} \times \mathsf{SS0}$ is true. Though, for readability's sake, we will prefer to rewrite a wff of the form $\neg\sigma = \tau$ as $\sigma \neq \tau$, so that last wff becomes $\mathsf{S0} + \mathsf{SS0} \neq \mathsf{SS0} \times \mathsf{SS0}$.

We will also give L_B the conditional connective.

6.2 The axioms and logic of Baby Arithmetic

The theory BA couched in this language L_B will come equipped with a classical deductive system to deal with negation, the conditional and identity. We can take the principles governing the connectives to be familiar. And to deal with identity, we need the principle that any sentence of the form $\tau = \tau$ is a logical truth, together with Leibniz's Law which allows us to intersubstitute identicals – in other words, given $\varphi(\tau)$ and either $\sigma = \tau$ or $\tau = \sigma$, we can infer $\varphi(\sigma)$. In illustrations, we'll set out proofs in a Fitch-like natural deduction format (because it is likely to be familiar, and is in any case easy to follow): nothing hangs on this choice of logical system.

Next, we want non-logical axioms governing the successor function. We want to capture the idea that, if we start from zero and repeatedly apply the successor function, we keep on getting further numbers – i.e. different numbers have different successors: contraposing, for any m, n, if $Sm = Sn$ then $m = n$. Further, zero isn't a successor, i.e. we never cycle back to zero: for any n, $0 \neq Sn$.

However, there are no quantifiers in L_B. So we can't directly express those general facts about the successor function inside the object language L_B. Rather, we have to employ *schemas* (i.e. general templates) and use the generalizing apparatus in our English metalanguage. So we say *any sentence that you get from one of the following schemas by substituting standard numerals for the place-holders 'ζ', 'ξ' is an axiom:*

Schema 1. $\mathsf{0} \neq \mathsf{S}\zeta$

Schema 2. $\mathsf{S}\zeta = \mathsf{S}\xi \;\rightarrow\; \zeta = \xi$

NB: These schemas are *not* axioms of BA; the Greek metavariables don't belong to the language L_B. It is, to repeat, *instances* of the schemas got by systematically replacing the placeholders with numerals – same placeholder, same replacement – which are the axioms.[1] We'll see some examples in a moment.

Next, we want non-logical axioms for addition that capture some key ideas underlying the school-room rules. So first we claim that adding zero to a number makes no difference: for any m, $m+0=m$. And next, adding a non-zero number Sn (i.e. $n+1$) to m is governed by the following rule: for any m, n, $m+Sn = S(m+n)$ – i.e. $m+(n+1) = (m+n)+1$. These two principles together tell us how to add zero to a given number m; and then adding one is defined as the successor of the result of adding zero; and then adding two is defined as the successor of the result of adding one; and so on up – thus defining adding n for any particular natural number n.

Because of L_B's lack of quantifiers, we again can't express all that directly inside L_B itself. We have to resort to schemas, and say that anything you get by substituting standard numerals for placeholders in one of the following schemas is an axiom – for short, *every numeral instance of these schemas is an axiom*:

Schema 3. $\zeta + 0 = \zeta$

Schema 4. $\zeta + \mathsf{S}\xi = \mathsf{S}(\zeta + \xi)$

We can similarly pin down the multiplication function by requiring that *every numeral instance of these schemas too is an axiom*:

Schema 5. $\zeta \times 0 = 0$

Schema 6. $\zeta \times \mathsf{S}\xi = (\zeta \times \xi) + \zeta$

Instances of Schema 5 tell us the result of multiplying by zero. Instances of Schema 6 with 'ξ' replaced by '0' define how to multiply by one in terms of first multiplying by zero and then applying the already-defined addition function. Once we know about multiplying by one, we can use another instance of Schema 6 – this time with 'ξ' replaced by '$\mathsf{S}0$' – to tell us how to multiply by two (multiply by one and then do some addition). And so on, thus defining multiplication for every number.

To summarize, then,

Defn. 19. BA*,* Baby Arithmetic*, is the theory whose language is L_B, whose logic comprises classical rules for negation and the conditional, together with identity rules, and whose non-logical axioms are every numeral instance of Schemas (1) to (6).*

So although BA's axioms fall into just six kinds, there are an infinite number of them – since *any* instance of our schemas counts as an axiom. However, although

[1] Fine print: here and below, it wouldn't actually make any difference to the strength of our theory if we allowed the placeholding metavariables to be systematically replaced by any terms, not just by standard numerals. But let's keep things simple.

Note that *here* we can't use free variables as placeholders – compare §4.2 fn.1. For L_B doesn't have any variables we can recruit for this duty.

it isn't *finitely* axiomatized, it is still an *effectively* axiomatized theory: given a candidate wff, it is clear we can effectively decide whether it is an instance of one of those six schemas and hence an axiom.[2]

A final remark. Suppose we had adopted versions of our everyday English numerals to denote numbers in a formal arithmetic. Then we would also need a whole bunch of additional axioms like $\mathsf{S\,zero} = \mathsf{one}$, $\mathsf{S\,one} = \mathsf{two}$, and so on, plus further school-room rules to deal with our base-ten notation. We have avoided this sort of complication by choosing to use our standard numerals to denote numbers in BA. So that's a non-trivial choice. We are already building into our system for denoting numbers something of the structure of number series, and it is this which enables our axioms for BA to be so simple. Which is all perfectly legitimate: but we should be aware that this *is* what we are doing.

6.3 Proofs of equations inside BA

Let's start with three brisk examples of how arithmetic can be done inside BA, breaking down some informal calculations into minimal steps. (These examples are decidedly unexciting: but arguing 'Here are some BA derivations of equations, and we can obviously generalize from these particular cases to get Theorem 9' will actually be more illuminating than giving an abstract general proof of our next theorem.)

First, let's show that $\mathsf{BA} \vdash 0 + \overline{2} = \overline{2}$. In other words, $0 + \mathsf{SS}0 = \mathsf{SS}0$ is a theorem – and note carefully, this wff *isn't* an instance of Schema 3.

1.	$0 + 0 = 0$	Axiom, instance of Schema 3
2.	$0 + \mathsf{S}0 = \mathsf{S}(0 + 0)$	Axiom, instance of Schema 4
3.	$0 + \mathsf{S}0 = \mathsf{S}0$	From 1, 2 by Leibniz's Law (LL)
4.	$0 + \mathsf{SS}0 = \mathsf{S}(0 + \mathsf{S}0)$	Axiom, instance of Schema 4
5.	$0 + \mathsf{SS}0 = \mathsf{SS}0$	From 3, 4 by LL

Similarly, we can prove $\overline{2} + \overline{2} = \overline{4}$, i.e. $\mathsf{SS}0 + \mathsf{SS}0 = \mathsf{SSSS}0$:

1.	$\mathsf{SS}0 + 0 = \mathsf{SS}0$	Axiom, instance of Schema 3
2.	$\mathsf{SS}0 + \mathsf{S}0 = \mathsf{S}(\mathsf{SS}0 + 0)$	Axiom, instance of Schema 4
3.	$\mathsf{SS}0 + \mathsf{S}0 = \mathsf{SSS}0$	From 1, 2 by LL
4.	$\mathsf{SS}0 + \mathsf{SS}0 = \mathsf{S}(\mathsf{SS}0 + \mathsf{S}0)$	Axiom, instance of Schema 4
5.	$\mathsf{SS}0 + \mathsf{SS}0 = \mathsf{SSSS}0$	From 3, 4 by LL

And now let's show that $\mathsf{BA} \vdash \overline{2} \times \overline{2} = \overline{4}$. In unabbreviated form, we need (rather laboriously!) to derive $\mathsf{SS}0 \times \mathsf{SS}0 = \mathsf{SSSS}0$:

[2]More fine print. We definitely want negation in our language of Baby Arithmetic, as we want to be able to formally express true inequalities such as $1 + 1 \neq 3$. But the use of the conditional is in fact optional. As should become clear, we could for our purposes trade in the schema that says that every wff of the form $\mathsf{S}\zeta = \mathsf{S}\xi \to \zeta = \xi$ is true for a corresponding inference rule that tells us that from a wff of the form $\mathsf{S}\zeta = \mathsf{S}\xi$ we can infer the corresponding wff of the form $\zeta = \xi$. Not having the conditional in play would make the proof of Theorem 13 one step simpler; but it would slightly obscure the point that the key move between Baby Arithmetic and Robinson Arithmetic is adding the quantifiers.

1.	$\mathsf{SS0} \times \mathsf{0} = \mathsf{0}$	Axiom, instance of Schema 5
2.	$\mathsf{SS0} \times \mathsf{S0} = (\mathsf{SS0} \times \mathsf{0}) + \mathsf{SS0}$	Axiom, instance of Schema 6
3.	$\mathsf{SS0} \times \mathsf{S0} = \mathsf{0} + \mathsf{SS0}$	From 1, 2 by LL
4.	$\mathsf{0} + \mathsf{SS0} = \mathsf{SS0}$	Derived as in first proof above
5.	$\mathsf{SS0} \times \mathsf{S0} = \mathsf{SS0}$	From 3, 4 by LL
6.	$\mathsf{SS0} \times \mathsf{SS0} = (\mathsf{SS0} \times \mathsf{S0}) + \mathsf{SS0}$	Axiom, instance of Schema 6
7.	$\mathsf{SS0} \times \mathsf{SS0} = \mathsf{SS0} + \mathsf{SS0}$	From 5, 6 by LL
8.	$\mathsf{SS0} + \mathsf{SS0} = \mathsf{SSSS0}$	Derived as in second proof above
9.	$\mathsf{SS0} \times \mathsf{SS0} = \mathsf{SSSS0}$	From 7, 8 by LL

OK: so now let's generalize. Suppose that for some other m we'd started instead from the Axiom $\overline{\mathsf{m}} + \mathsf{0} = \overline{\mathsf{m}}$, another instance of Schema 3. Then by similar steps as for the first two proofs, we can derive $\overline{\mathsf{m}} + \mathsf{SS0} = \mathsf{SS}\overline{\mathsf{m}}$, i.e. $\overline{\mathsf{m}} + \overline{\mathsf{2}} = \overline{\mathsf{m + 2}}$ (here, $\overline{\mathsf{m + 2}}$ of course stands in for the standard numeral for $m + 2$).

And then, generalizing further, if we keep extending the same proof idea with a few more steps cut to the same pattern, we can get BA to show $\overline{\mathsf{m}} + \overline{\mathsf{3}} = \overline{\mathsf{m + 3}}$, and $\overline{\mathsf{m}} + \overline{\mathsf{4}} = \overline{\mathsf{m + 4}}$, and so on. In fact, for any m, n, $\mathsf{BA} \vdash \overline{\mathsf{m}} + \overline{\mathsf{n}} = \overline{\mathsf{m + n}}$.

Next, looking at our third sample proof, we see that we'll be able to similarly prove $\overline{\mathsf{m}} \times \overline{\mathsf{2}} = \overline{\mathsf{m \times 2}}$ for any m. And then, generalizing further, if we keep extending the same proof idea with more steps cut to the same pattern, we can prove $\overline{\mathsf{m}} \times \overline{\mathsf{3}} = \overline{\mathsf{m \times 3}}$, and $\overline{\mathsf{m}} \times \overline{\mathsf{4}} = \overline{\mathsf{m \times 4}}$, and so on. In fact, for any m, n, $\mathsf{BA} \vdash \overline{\mathsf{m}} \times \overline{\mathsf{n}} = \overline{\mathsf{m \times n}}$.

We can now generalize a step further: BA can correctly evaluate not just the simplest terms but *all* terms of its language. That is to say,

Theorem 9. *Suppose τ is a term of L_B and suppose the value of τ on the intended interpretation of the symbols is t. Then* $\mathsf{BA} \vdash \tau = \overline{\mathsf{t}}$.

Why so? Well, let's take a very simple example and then draw a general moral. Suppose we want to show e.g. that $(\overline{\mathsf{2}} + \overline{\mathsf{3}}) \times (\overline{\mathsf{2}} \times \overline{\mathsf{2}}) = \overline{\mathsf{20}}$ – you'll forgive me for not writing out '$\overline{\mathsf{20}}$' in basic notation with its twenty occurrences of 'S'! Then we can proceed as follows, again arguing in BA:

1.	$(\overline{\mathsf{2}} + \overline{\mathsf{3}}) \times (\overline{\mathsf{2}} \times \overline{\mathsf{2}}) = (\overline{\mathsf{2}} + \overline{\mathsf{3}}) \times (\overline{\mathsf{2}} \times \overline{\mathsf{2}})$	Identity law
2.	$\overline{\mathsf{2}} + \overline{\mathsf{3}} = \overline{\mathsf{5}}$	BA can do simple addition
3.	$(\overline{\mathsf{2}} + \overline{\mathsf{3}}) \times (\overline{\mathsf{2}} \times \overline{\mathsf{2}}) = \overline{\mathsf{5}} \times (\overline{\mathsf{2}} \times \overline{\mathsf{2}})$	From 1, 2 by LL
4.	$\overline{\mathsf{2}} \times \overline{\mathsf{2}} = \overline{\mathsf{4}}$	BA can do simple multiplication
5.	$(\overline{\mathsf{2}} + \overline{\mathsf{3}}) \times (\overline{\mathsf{2}} \times \overline{\mathsf{2}}) = \overline{\mathsf{5}} \times \overline{\mathsf{4}}$	From 3, 4 by LL
6.	$\overline{\mathsf{5}} \times \overline{\mathsf{4}} = \overline{\mathsf{20}}$	BA can do simple multiplication
7.	$(\overline{\mathsf{2}} + \overline{\mathsf{3}}) \times (\overline{\mathsf{2}} \times \overline{\mathsf{2}}) = \overline{\mathsf{20}}$	From 5, 6 using LL

What we do here is 'evaluate' the complex formula on the right 'from the inside out', reducing the complexity of what's on the right at each stage, and hence eventually equating the complex formula on the left with a standard numeral on the right. Evidently, we can always do this trick, whatever complex formula we start from.

From this last result, we can immediately deduce

Theorem 10. *If $\sigma = \tau$ is a true equation, then* $\mathsf{BA} \vdash \sigma = \tau$.

Proof. If $\sigma = \tau$ is true, then σ and τ must evaluate to the same number n. Hence by Theorem 9, we have both $\mathsf{BA} \vdash \sigma = \bar{\mathsf{n}}$ and $\mathsf{BA} \vdash \tau = \bar{\mathsf{n}}$. From which it immediately follows that $\mathsf{BA} \vdash \sigma = \tau$ by Leibniz's Law. ⊠

6.4 Proofs of inequations inside BA

Next, we note that BA knows that different standard numerals are indeed not equal. For example, here's a BA proof of $\bar{4} \neq \bar{2}$:

1.	$\mathsf{SSSS0} = \mathsf{SS0}$	Supposition
2.	$\mathsf{SSSS0} = \mathsf{SS0} \to \mathsf{SSS0} = \mathsf{S0}$	Axiom, instance of Schema 2
3.	$\mathsf{SSS0} = \mathsf{S0}$	From 1, 2 by Modus Ponens
4.	$\mathsf{SSS0} = \mathsf{S0} \to \mathsf{SS0} = 0$	Axiom, instance of Schema 2
5.	$\mathsf{SS0} = 0$	From 3, 4 by Modus Ponens
6.	$0 \neq \mathsf{SS0}$	Axiom, instance of Schema 1
7.	Contradiction!	From 5, 6 and identity rules
8.	$\mathsf{SSSS0} \neq \mathsf{SS0}$	From 1 to 7, by Reductio ad Absurdum.

And a little reflection on this illustrative proof should now convince you that in general

Theorem 11. *If s and t are distinct numbers, then* $\mathsf{BA} \vdash \bar{\mathsf{s}} \neq \bar{\mathsf{t}}$.

And that immediately gives us a companion result to Theorem 10:

Theorem 12. *If $\sigma = \tau$ is a false equation, then* $\mathsf{BA} \vdash \sigma \neq \tau$.

Proof. If $\sigma = \tau$ is false, then σ will evaluate to s and τ will evaluate to t where $s \neq t$. So by Theorem 10, we have both $\mathsf{BA} \vdash \sigma = \bar{\mathsf{s}}$ and $\mathsf{BA} \vdash \tau = \bar{\mathsf{t}}$, and by Theorem 11 $\mathsf{BA} \vdash \bar{\mathsf{s}} \neq \bar{\mathsf{t}}$. So two applications of Leibniz's Law give us the desired result, $\mathsf{BA} \vdash \sigma \neq \tau$. ⊠

6.5 BA is a sound and negation-complete theory of the truths of L_B

Theorems 10 and 12 tell us that, as far as the atomic wffs of L_B are concerned (i.e. the equations $\sigma = \tau$), the true ones are provable in BA and the false ones are refutable. So we could say that BA is negation complete for equations.

And we can now easily show that BA is negation complete for *all* wffs of L_B, by a simple appeal to the completeness of truth-functional propositional logic.

A wff φ of L_B is built using negation and other truth-functional connectives from the atoms $\alpha_1, \alpha_2, \ldots, \alpha_n$. Consider any assignment V of truth-values to those atoms. Let α_i^V be α_i if that is true on V, and $\neg\alpha_i$ otherwise. Similarly, let φ^V be φ if that is true on V, and $\neg\varphi$ otherwise.

So by the definition of the α_i^V and φ^V, the one and only way of making the α_i^V all true together (i.e. valuation V) makes φ^V true. Hence $\alpha_1^V, \alpha_2^V, \ldots, \alpha_n^V$ tautologically entail φ^V. Hence by the completeness theorem for your favourite propositional logic PL, $\alpha_1^V, \alpha_2^V, \ldots, \alpha_n^V \vdash_{PL} \varphi^V$. We can now use this to show:

Theorem 13. BA *is a sound effectively axiomatized theory which is negation complete.*

Proof. BA is evidently a sound theory – all its axioms are trivial arithmetical truths, and its logic is truth-preserving, so all its theorems are true. It is also effectively axiomatized.

Now take any L_B sentence φ. This is a truth-functional combination of atomic formulae α_i (i.e. equations). Consider the valuation V which gives these atoms (equations) their arithmetically correct values. Then Theorems 10 and 12 tell us that $\mathsf{BA} \vdash \alpha_i^V$ for each equation α_i. And BA contains a propositional logic PL. So, by the fact just noted, BA proves φ^V – i.e. it proves whichever of φ and $\neg\varphi$ is the true one. So BA is negation complete. ⊠

"Hold on! I thought we couldn't have a sound effectively axiomatized theory of arithmetic which is negation complete." No. Theorem 1 didn't say *that*: it said we couldn't have a sound, negation-complete, effectively axiomatized theory which contains what we called the language of basic arithmetic – and *that* language allows us to quantify over numbers. By contrast, L_B is quantifier-free. This language only allows us to express facts about adding and multiplying particular numbers (it can't express numerical generalizations). That's why it can be complete.

"Ah. So having quantifiers in a theory's language can make all the difference?" Yes!

6.6 The language L_A

So far that is all very straightforward, but also rather unexciting.[3] The reason that Baby Arithmetic manages to prove every correct claim that it can express – and is therefore negation complete by our Defn. 7 – is that it can't express very much. In particular, as we just stressed, it can't express any generalizations at all. And so the obvious way to beef up BA into something more expressively competent is to restore the familiar apparatus of quantifiers and variables. That's what we'll do next.

First, then, we define the interpreted *first-order language of basic arithmetic* L_A. We will keep the same non-logical vocabulary as in L_B: so there is still just a single non-logical constant denoting zero, plus the three function-symbols, $\mathsf{S}, +, \times$, still expressing successor, addition and multiplication. But now we allow ourselves the full expressive resources of first-order logic; so we now have the propositional connectives plus the usual supply of quantifiers and variables to

[3] Mathematically unexciting, anyway. But there is perhaps some philosophical interest. For we might reasonably suppose that the axiom schemas of BA at least partially encapsulate the meanings of the symbols for zero and for the successor, addition and multiplication functions – they partially define what we are talking about. So it is consequently really rather tempting to be a logicist at least about the arithmetic truths proved by BA, regarding them as truths of logic-plus-definitions. And this success might encourage us to pursue some more ambitious form of logicism (see §2.5).

express generality, as well as the built-in identity predicate. We fix the domain of the quantifiers to be the natural numbers. The result is the language L_A: and this should look familiar – it is the least ambitious language which 'contains the language of basic arithmetic' in the sense of Defn. 10.

6.7 Robinson Arithmetic, Q

With this richer formal language available, we can define *Robinson Arithmetic*, commonly denoted simply 'Q'.[4] This is a theory built in the language L_A. It is equipped with a full proof system for first-order classical logic. And for its non-logical axioms, now that we have the quantifiers available to express generality, we can replace each of BA's metalinguistic schemas (specifying an infinite number of formal axioms governing particular numbers) by a single generalized Axiom expressed inside L_A itself.

For example, we can replace the first two schemas governing the successor function by the following:

Axiom 1. $\forall x(0 \neq Sx)$

Axiom 2. $\forall x \forall y(Sx = Sy \to x = y)$

Obviously, each instance of our earlier Baby Arithmetic Schemas 1 and 2 can be deduced from the corresponding Robinson Arithmetic Axiom by instantiating the quantifiers with numerals.

These Axioms tell us that zero isn't a successor, but they don't explicitly rule out there being *other* objects that aren't successors cluttering up the domain of quantification. We didn't need to fuss about this before, because by construction BA can only talk about the numbers represented by standard numerals in the sequence '$0, S0, SS0, \ldots$'. But now we have the quantifiers in play. And these quantifiers are intended to run over the natural numbers, i.e. over zero and its successors.

So let's add an axiom which says that, other than zero, every number is indeed a successor:

Axiom 3. $\forall x(x \neq 0 \ \to \ \exists y(x = Sy))$

Next, we can similarly replace our previous schemas for addition and multiplication by universally quantified Axioms in the obvious way:

Axiom 4. $\forall x(x + 0 = x)$

Axiom 5. $\forall x \forall y(x + Sy = S(x + y))$

Axiom 6. $\forall x(x \times 0 = 0)$

Axiom 7. $\forall x \forall y(x \times Sy = (x \times y) + x)$

Again, each of these Q axioms entails all the instances of BA's corresponding schema.

[4]The expected 'R' is in fact the name given to a different Robinsonian arithmetic – see fn.7.

In sum, then:

Defn. 20. *The formal theory with language L_A, Axioms 1 to 7, plus a classical first-order logic, is standardly called* Robinson Arithmetic, *or simply* Q.

Since any BA axiom can be derived from one of our new Q Axioms, anything that can be proved in BA can be proved in Q.

6.8 Robinson Arithmetic is not complete

Like BA, Q too is an effectively axiomatized sound theory. Its axioms are all true; and its logic is truth-preserving; so its derivations are genuine proofs in the intuitive sense of demonstrations of truth. Every theorem of Q is a true L_A wff, then. But just which truths of L_A are theorems of Q?

Well, on the positive side,

Theorem 14. Q *correctly decides every quantifier-free L_A sentence. In other words,* $\mathsf{Q} \vdash \varphi$ *if the quantifier-free wff φ is true, and* $\mathsf{Q} \vdash \neg\varphi$ *if the quantifier-free wff φ is false.*

Proof. We know that Q (like BA) will correctly decide every atomic wff, i.e. correctly decide every equation between terms. And as in our proof of Theorem 13, it follows that Q must then correctly decide every wff built up from those atoms using just the truth-functional propositional connectives. ⊠

So far, so good. However, there are very simple true *quantified* sentences that Q can't prove.

For example, while Q can prove any particular wff of the form $0 + \overline{n} = \overline{n}$, *it can't prove the corresponding universal generalization*:

Theorem 15. $\mathsf{Q} \nvdash \forall x(0 + x = x)$.

Proof Since Q is a theory with a standard first-order theory, for any L_A-sentence φ, $\mathsf{Q} \vdash \varphi$ only if $\mathsf{Q} \vDash \varphi$ (that's just the soundness theorem for first-order logic).

Put $\mathsf{U} =_{\text{def}} \forall x(0 + x = x)$. Then one way of showing that $\mathsf{Q} \nvdash \mathsf{U}$ is to show that $\mathsf{Q} \nvDash \mathsf{U}$: and we can show *that* by producing a countermodel to the entailment – i.e. by finding an interpretation (a deviant, unintended, 'non-standard', re-interpretation) for L_A's wffs which makes Q's axioms true-on-that-interpretation but which makes U false.

So here goes: take the domain of our deviant, unintended, re-interpretation to be the set N^* which comprises the natural numbers but with two other 'rogue' elements a and b added (these could be e.g. Kurt Gödel and his friend Albert Einstein – but any other pair of distinct non-numbers will do). Let '0' still refer to zero. And take 'S' now to pick out the successor* function S^* which is defined as follows: $S^*n = Sn$ for any natural number in the domain, while for our rogue elements $S^*a = a$, and $S^*b = b$. It is very easy to check that Axioms 1 to 3 are still true on this deviant interpretation. Zero is still not a successor. Different elements have different successors. And every non-zero element is again a successor (perhaps a self-successor! – though not necessarily an eventual successor of zero).

We now need to extend this interpretation to re-interpret the function-symbol '+'. Suppose we take this to pick out addition*, where $m +^* n = m + n$ for any natural numbers m, n in the domain, while $a +^* n = a$ and $b +^* n = b$. Further, for any x (whether number or rogue element), $x +^* a = b$ and $x +^* b = a$. If you prefer that in a table, then read off *row* $+^*$ *column* here:

$+^*$	n	a	b
m	$m + n$	b	a
a	a	b	a
b	b	b	a

It is again easily checked that interpreting '+' in Q as addition* still makes Axioms 4 and 5 true.[5] And note that we have e.g. $0 +^* a \neq a$, so on this interpretation U, i.e. $\forall x(0 + x = x)$, fails.

We are not quite done, however, as we still need to show that we can give a co-ordinate re-interpretation of '×' in Q by some deviant multiplication* function. We can leave it as an exercise to fill in suitable details.

Then, with the details filled in, we will have in short an overall interpretation which makes the axioms of Q true and U false. So $\mathsf{Q} \nvdash \mathsf{U}$ ⊠

Theorem 16. Q *is negation incomplete.*

Proof. We've just shown that Q can't prove U. But obviously, Q can't prove ¬U either. Revert to the standard interpretation built into L_A. All Q's theorems are true but ¬U is false on that interpretation. So ¬U can't be a theorem. Hence U is formally undecidable in Q. ⊠

Of course, we've already announced that Gödel's incompleteness theorem is going to prove that *no* sound axiomatized theory whose language is at least as rich as L_A can be negation complete – that was Theorem 1. But we don't need to invoke anything remotely as elaborate as Gödel's arguments to see that Q is negation incomplete. Q is, so to speak, *boringly* incomplete.

6.9 Statements of order in Robinson Arithmetic

We found a deviant interpretation of Q's axioms by exploiting the fact that, while Axiom 3 ensures every object other than zero is a successor of something, that axiom still allows 'stray objects' which aren't eventual successors of zero. So obviously we'll want to add more axioms to Q in the hope of pinning down

[5] In headline terms: For Axiom 4, we note that adding* zero on the right always has no effect. For Axiom 5, just consider cases. (i) $m +^* S^*n = m + Sn = S(m + n) = S^*(m +^* n)$ for 'ordinary' numbers m, n in the domain. (ii) $a + S^*n = a = S^*a = S^*(a +^* n)$, for 'ordinary' n. Likewise, (iii) $b + S^*n = S^*(b +^* n)$. (iv) $x +^* S^*a = x + a = b = S^*b = S^*(x +^* a)$, for any x in the domain. (v) Similarly, $x +^* S^*b = S^*(x +^* b)$. Which covers every possibility.

the structure of the natural numbers and eliminating 'strays'. That's our task in the next chapter. But let's stick with Robinson Arithmetic for a bit longer. For as we will note in the next section, weak though it is, Q has a property which makes it of considerable interest.

And as a warm-up in this section, we will prove a result that will later be useful.

Theorem 17. *In Robinson Arithmetic, the* less-than-or-equal-to *relation is not just expressed but captured by the wff* $\exists \mathsf{v}(\mathsf{v} + \mathsf{x} = \mathsf{y})$.

Proof. It is obvious that the wff expresses the relation. So – recalling the definition of capturing in §5.2, Defn. 17 – what we need to show is that, for any particular pair of numbers, m, n, if $m \leq n$, then $\mathsf{Q} \vdash \exists \mathsf{v}(\mathsf{v} + \overline{\mathsf{m}} = \overline{\mathsf{n}})$, and if $m > n$, then $\mathsf{Q} \vdash \neg\exists \mathsf{v}(\mathsf{v} + \overline{\mathsf{m}} = \overline{\mathsf{n}})$.

Suppose $m \leq n$, so for some $k \geq 0$, $k + m = n$. Q can prove everything BA proves and hence, in particular, can prove every true addition equation. So we have $\mathsf{Q} \vdash \overline{\mathsf{k}} + \overline{\mathsf{m}} = \overline{\mathsf{n}}$. But then $\exists \mathsf{v}(\mathsf{v} + \overline{\mathsf{m}} = \overline{\mathsf{n}})$ follows by existential quantifier introduction. Therefore $\mathsf{Q} \vdash \exists \mathsf{v}(\mathsf{v} + \overline{\mathsf{m}} = \overline{\mathsf{n}})$, as was to be shown.

Suppose alternatively $m > n$. We need to show $\mathsf{Q} \vdash \neg\exists \mathsf{v}(\mathsf{v} + \overline{\mathsf{m}} = \overline{\mathsf{n}})$. We'll first demonstrate this in the case where $m = 2$, $n = 1$, using a Fitch-style proof system. For brevity we will omit statements of Q's axioms and some other trivial steps; we drop unnecessary brackets too.

1.	$\exists \mathsf{v}(\mathsf{v} + \mathsf{SS0} = \mathsf{S0})$	Supposition
2.	$\mathsf{a} + \mathsf{SS0} = \mathsf{S0}$	Supposition
3.	$\mathsf{a} + \mathsf{SS0} = \mathsf{S}(\mathsf{a} + \mathsf{S0})$	From Axiom 5
4.	$\mathsf{S}(\mathsf{a} + \mathsf{S0}) = \mathsf{S0}$	From 2, 3 by LL
5.	$\mathsf{a} + \mathsf{S0} = \mathsf{S}(\mathsf{a} + 0)$	From Axiom 5
6.	$\mathsf{SS}(\mathsf{a} + 0) = \mathsf{S0}$	From 4, 5 by LL
7.	$\mathsf{a} + 0 = \mathsf{a}$	From Axiom 4
8.	$\mathsf{SSa} = \mathsf{S0}$	From 6, 7 by LL
9.	$\mathsf{SSa} = \mathsf{S0} \to \mathsf{Sa} = 0$	From Axiom 2
10.	$\mathsf{Sa} = 0$	From 8, 9 by Modus Ponens
11.	$0 = \mathsf{Sa}$	From 10
12.	$0 \neq \mathsf{Sa}$	From Axiom 1
13.	Contradiction!	From 11, 12
14.	Contradiction!	$\exists$E 1, 2–13
15.	$\neg\exists \mathsf{v}(\mathsf{v} + \mathsf{SS0} = \mathsf{S0})$	From 1–14 by Reductio

The only step to explain may be at line (14) where we use a version of the Existential Elimination rule: if the temporary supposition $\varphi(\mathsf{a})$ leads to contradiction, for arbitrary a, then $\exists \mathsf{v}\varphi(\mathsf{v})$ must lead to contradiction.

And having constructed a proof for the case $m = 2$, $n = 1$, inspection reveals that we can use the same general pattern of argument to show $\mathsf{Q} \vdash \neg\exists \mathsf{v}(\mathsf{v} + \overline{\mathsf{m}} = \overline{\mathsf{n}})$ whenever $m > n$. So we are done. ⊠

Given the theorem we have just proved, we can sensibly add the standard symbol '$\leq$' to L_A, the language of Q, defined so that for any L_A terms – including of course variables – $\sigma \leq \tau$ is just short for $\exists \mathsf{v}(\mathsf{v} + \sigma = \tau)$.[6] And then Q will be able to prove at least the expected facts about the less-than-or-equals relations among quantifier-free terms.

Note, by the way, that some presentations treat '$\leq$' as a primitive symbol built into our formal theories like Q from the start, governed by its own additional axiom(s). But nothing important hangs on the difference between that approach and our policy of introducing the symbol by definition.

And of course, nothing hangs either on our policy of introducing '$\leq$' as our basic symbol rather than '$<$', which could have been defined by $\sigma < \tau =_{\mathrm{def}} \exists \mathsf{v}(\mathsf{Sv} + \sigma = \tau)$.

6.10 Why Robinson Arithmetic is interesting

Given it can't even prove $\forall \mathsf{x}(0 + \mathsf{x} = \mathsf{x})$, Q is evidently a *very* weak theory of arithmetic. Even so, Q does have some very interesting features.

As we saw, it can capture the particular decidable relation that obtains when one number is at least as big as another. And in fact, we can now announce a surprisingly sweeping general result:

Theorem 18. Q *can capture* all *effectively decidable numerical properties (and relations) – hence it is sufficiently strong in the sense of Defn 18.*

That might initially seem *very* unexpected, given Q's weakness. But remember, 'sufficient strength' was defined as a matter of being able to *case-by-case* prove enough wffs about decidable properties of individual numbers. It turns out that Q's hopeless weakness at proving *generalizations* doesn't stop it from proving enough facts about *particular* numbers.

So that's why Q is especially interesting – it is one of the very weakest arithmetics which is sufficiently strong, and it was isolated by Raphael Robinson in 1952 for just that reason.[7] So Q is one of the weakest arithmetics for which Gödelian proofs of incompleteness can be run. Suppose, then, that a theory is formally axiomatized, consistent and can prove everything Q can prove (surely very weak requirements). Then what we've just announced and promised can be proved is that any such theory will be sufficiently strong. And therefore e.g. Theorem 8 will apply – any such theory will be incomplete.

However, we can only prove the announced Theorem 18 that Q *does* have sufficient strength if and when we have a quite general theory of effective decidability to hand. And as we said at the end of the previous chapter, we don't want to get embroiled yet in developing that theory. So what we *will* be proving quite soon (in Chapter **??**) is a somewhat weaker claim about Q. We'll show that it can

[6] Fine print: strictly, we need to choose the quantified variable to avoid any clash of variables with the substituted terms. We won't fuss about that.

[7] We can't say that Q is *the* weakest sufficiently strong arithmetic. Robinson also isolated another very weak but sufficiently strong arithmetic R which neither contains nor is contained in Q. But Robinson's other theory isn't finitely axiomatized, so it is usual to focus on the prettier and finitely axiomatized Q.

capture all so-called 'primitive recursive' properties and relations, where these form a large and very important subclass of the effectively decidable ones. This major theorem will be a crucial load-bearing part of our proofs of various Gödel style incompleteness theorems: it means that Q gives us 'the modest amount of arithmetic' needed for a version of Theorem 2.

But before we get round to showing all that, we are first going to take a look at a *much* richer arithmetic than Q, namely PA.

7 First-order Peano Arithmetic

The previous chapter introduced two weak theories of arithmetic, BA and Q. In this chapter – jumping over a whole family of intermediate-strength theories – we introduce a *much* richer first-order theory of arithmetic, PA. It's what you get by adding a generous supply of induction axioms to Q.

7.1 Mathematical induction: the very idea

Here is the basic idea we need (and throughout, take 'number' to mean 'natural number'):

> Whatever numerical property we take, if (i) zero has this property, and also (ii) the property is passed down from any number n which has it to its successor Sn, then it follows that (iii) the property is had by *all* numbers.

This is the key *principle of mathematical induction*, used in very many informal proofs of arithmetical generalizations.[1]

The principle is plainly a sound one, guaranteed by the structure of the number series. Why? Suppose (i) 0 has property P and also (ii) for any n, if n has P so does Sn. Since 0 has P, we can use (ii) to deduce $S0$ has P. Since $S0$ has P, we can use (ii) to deduce that $SS0$ has P. Then, by appeal to (ii) again, it follows that $SSS0$ has P. And so on. Hence (iii) property P will in this way percolate down to any given number – since you can get to any natural number by starting from zero and repeatedly applying the successor function (there are no stray natural numbers, lying outside that sequence).

7.2 The induction axiom, the induction schema

The intuitive idea, to repeat, is that for any property of numbers, if zero has it and if it is passed from one number to the next, then all numbers have it. Putting it this way involves generalizing over properties of numbers. So to frame a corresponding formal induction principle, we might naturally want to use a formalized

[1] I'll assume this is familiar. If not, you can start by looking at the Wikipedia article on 'Mathematical induction', or (better!) look at the chapter with the same title in Daniel J. Velleman's *How to Prove It* (CUP).

language that likewise allows us to generalize over properties of numbers. And the obvious option is to adopt a language with *second-order* quantifiers.

In other words, we will want a language that not only has first-order quantifiers of the familiar sort, in this case running over the natural numbers, but also has a second type of quantifier which runs over all properties-of-numbers. In such a language, we can then state a *second-order Induction Axiom* as follows:

$$\forall \mathsf{X}(\{\mathsf{X0} \land \forall \mathsf{x}(\mathsf{Xx} \to \mathsf{XSx})\} \to \forall \mathsf{xXx})$$

You can read this as saying 'For any property X, given both that 0 has X and also that if any number has X so does its successor, then *every* number has property X.'[2]

Natural though this second-order axiom might be, we will however be focusing on formal theories whose logical apparatus involves only first-order quantification over numbers. In particular, we will be looking at theories which, like Q, are framed in the language L_A.

Why? This isn't due to some perverse desire to work with one hand tied behind our backs. It's because there are troublesome issues about second-order logic. For a start, there are technical issues: second-order logic's consequence relation (at least on the natural understanding) can't be captured in a nice formalizable logical system: hence theories using a full second-order logic aren't effectively axiomatizable. And then there are more philosophical issues: just how well do we really understand the intuitive idea of quantifying over 'all properties of numbers'? Is that really a determinate totality which we can quantify over? We don't want to tangle with such worries here and now.

But if we don't have second-order quantifiers available to range over properties of numbers, how can we handle arithmetical induction in a first-order setting?

Recall how we dealt with a similar problem in the context of Baby Arithmetic. The language L_B lacks first-order quantifiers; so we couldn't use a single quantified wff to formalize the claim that, for every number n, $n + 0 = n$. Instead, we had to adopt a whole package of unquantified axioms, with each numeral instance of the schema $\zeta + 0 = \zeta$ counting as a separate axiom. We'll now do something analogous here.

The language L_A lacks second-order quantifiers; so we can't write down a single second-order Axiom which directly formalizes the intuitive induction principle. Instead, we will have to adopt a whole package of axioms (which involve only first-order quantifiers) by saying that for each suitable wff $\varphi(\mathsf{x})$, the corresponding instance of the *first-order Induction Schema*

$$\{\varphi(\mathsf{0}) \land \forall \mathsf{x}(\varphi(\mathsf{x}) \to \varphi(\mathsf{Sx}))\} \to \forall \mathsf{x}\varphi(\mathsf{x})$$

is to count as an axiom.[3] Or at least, that is the basic idea. The next section elaborates.

[2] Conventionally, we use upper case letters for predicates; correspondingly, we use upper case letters for variables which can occupy predicate positions.

[3] Equivalently, we could adopt a rule of inference to the effect that, given $\varphi(\mathsf{0})$ and $\forall \mathsf{x}(\varphi(\mathsf{x}) \to \varphi(\mathsf{Sx}))$, we can deduce $\forall \mathsf{x}\varphi(\mathsf{x})$.

7.3 Being generous with induction

Suppose then that we are starting from the first-order arithmetic Q, and aim to build a richer theory in its language L_A by adding instances of the Induction Schema as additional axioms. But what makes for a 'suitable' predicate-expression $\varphi(\mathsf{x})$ for use in an instance of the Schema?

Well, consider *any* open wff $\varphi(\mathsf{x})$ of L_A with one free variable. This will be built up from no more than the constant term '0', the familiar successor, addition and multiplication functions, plus the identity relation and other logical apparatus. Therefore – you might very well suppose – it ought to be a perfectly determinate matter, for each n, whether $\varphi(\overline{\mathsf{n}})$ is true or not. In other words, $\varphi(\mathsf{x})$ ought to express a perfectly determinate arithmetical property of numbers (even if, in the general case, we can't always effectively decide whether a given number n has the property or not). But if $\varphi(\mathsf{x})$ does express a perfectly determinate arithmetical property P, then – by the intuitive argument of §7.1 – we should be able to argue by induction that if zero has P, and if P always gets passed down from a number to its successor, then all numbers have property P. *So why not be generous and allow any open one-variable L_A-wff $\varphi(\mathsf{x})$ to count as suitable for substitution in the Induction Schema?*

But now note that we will also want to make use of instances of the Induction Schema where $\varphi(\mathsf{x})$ has further variables dangling free.

Why? Consider informal mathematical argumentation again. We often want to establish universally generalized *relational* claims about numbers, and might then need to use an argument of the following overall shape:

> Pick an arbitrary number a. Then (i) 0 is R to a. And (ii) for any m, if m is R to a, so is Sm. Hence by induction, (iii) for any m, m is R to a. But a was arbitrary. So we can conclude that m is R to n for *any* numbers m, n.

The core part of the argument here is an induction for, so to speak, the property of *being-R-to-a* (where 'a' serves as a parameter, a temporary name for an arbitrarily selected number). So now the question is: how could we frame a formal analogue for the inductive argument here?

The obvious move is to allow instances of the induction schema for a wff $\varphi(\mathsf{x}, \mathsf{y})$ with the free variable 'y' serving as a parameter.[4]

Then, if you don't mind the axioms of a theory having free variables, we can simply say (in the same generous spirit as before) that each instance of the induction schema – now including those instances with variables dangling free in φ – is to be an axiom. If you do mind, then we can instead say that it is the *closure* of each instance of the induction schema which is an axiom (where, of course, the closure of a wff is what you get by universally quantifying any dangling free variables). I somewhat prefer the second line, because we can then

[4]Fine print: If your formal logic insists on typographically distinguishing free-variables-used-parametrically from bound variables, you'd instead use the likes of $\varphi(\mathsf{x}, \mathsf{a})$.

still think of a sound arithmetical theory as always having true axioms. But we needn't really fuss over which line to take.

7.4 First-order Peano Arithmetic introduced

Suppose then that we are generous with induction and agree that *any* open wff of L_A is suitable for use in an instance of the induction schema. This means moving on from Q, and jumping over a range of possible intermediate theories, to adopt the much richer theory of arithmetic that we can briskly define as follows:

Defn. 21. PA – First-order Peano Arithmetic[5] – *is the theory with a standard first-order logic whose language is* L_A *and whose axioms are those of* Q *plus (the closures of)* all *instances of the Induction Schema that can be constructed from open wffs of* L_A.

Like BA then, PA as presented here has an infinite number of axioms. However that's fine: it is plainly still decidable whether any given wff has the right shape to be one of the new axioms, so this is still a legitimate formalized theory.

To help fix ideas, then, let's have three easy examples of what we can formally prove using induction.

(a) First, we'll check that we have plugged the particular gap we noted in Q. Recall: while Q can prove each separate instance of $0 + \overline{n} = \overline{n}$, it very feebly can't prove $\forall x(0 + x = x)$. However, PA can.

How? We just put $0 + x = x$ for $\varphi(x)$, prove $\varphi(0)$, prove $\forall x(\varphi(x) \to \varphi(Sx))$, and use induction to conclude $\forall x\varphi(x)$. Spelling that out in tediously plodding detail:

1.	$0 + 0 = 0$	Instance of Q's Axiom 4
2.	$\quad 0 + a = a$	Supposition
3.	$\quad S(0 + a) = Sa$	From 2 by the identity laws
4.	$\quad 0 + Sa = S(0 + a)$	Instance of Q's Axiom 5
5.	$\quad 0 + Sa = Sa$	From 3, 4
6.	$0 + a = a \to 0 + Sa = Sa$	From 2–5 by Conditional Proof
7.	$\forall x(0 + x = x \to 0 + Sx = Sx)$	From 6, since a was arbitrary.
8.	$0 + 0 = 0 \land \forall x(0 + x = x \to 0 + Sx = Sx)$	From 1, 8
9.	$\{0 + 0 = 0 \land \forall x(0 + x = x \to 0 + Sx = Sx)\} \to \forall x(0 + x = x)$	Instance of Induction Schema
10.	$\forall x(0 + x = x)$	From 8, 9 by Modus Ponens

(b) For our next example, take $\varphi(x)$ to be $x \neq Sx$. Then PA trivially proves $\varphi(0)$ because that's Q's Axiom 1. PA also proves $\forall x(\varphi(x) \to \varphi(Sx))$ by contraposing Axiom 2. And then an induction axiom tells us that if we have both $\varphi(0)$ and

[5]The name is conventional. Giuseppe Peano did publish a list of axioms for arithmetic in 1889. But they weren't first-order, only explicitly governed the successor relation, and – as Peano acknowledged – had already been stated by Richard Dedekind.

$\forall x(\varphi(x) \to \varphi(Sx))$ we can deduce $\forall x\varphi(x)$, i.e. $\forall x\, x \neq Sx$, i.e. no number is a self-successor.

Simple! Yet this trivial little result is worth noting when we recall our deviant interpretation which made the axioms of Q true while making $\forall x(0 + x = x)$ false: that interpretation featured Kurt Gödel himself added to the domain as a rogue self-successor. A bit of induction, however, rules out self-successors.

(c) A third observation. PA allows, in particular, induction for the formula $\varphi(x) =_{\text{def}} (x \neq 0 \;\to\; \exists y(x = Sy))$.

But now note that the corresponding $\varphi(0)$ is a trivial theorem. $\forall x\varphi(Sx)$ is an equally trivial theorem (why?), and that logically entails $\forall x(\varphi(x) \to \varphi(Sx))$. So we can use an instance of the Induction Schema inside PA to derive $\forall x\varphi(x)$.

However, that's just Axiom 3 of Q. So our initial presentation of PA – as explicitly having all the Axioms of Q – involves a certain redundancy.

7.5 Summary overview of PA

Given its very natural motivation, PA is the benchmark axiomatized first-order theory of basic arithmetic. It is probably worth pausing, then, to bring together all the elements of its specification in one place.

First, to repeat, the *language* of PA is L_A, a first-order language whose non-logical vocabulary comprises just the constant '0', the one-place function symbol 'S', and the two-place function symbols '+', '×'. The built-in interpretation for L_A gives those symbols their familiar interpretation in elementary arithmetic and takes the quantifiers to run over the natural numbers.

Second, PA's deductive *proof system* is some standard version of classical first-order logic with identity. The differences between various presentations of the logic of course don't make a difference to what sentences can be proved in PA. We just sketched a proof using a Fitch-style system. Though, as we will see, it will soon be convenient to fix officially on a linear Hilbert-style axiomatic system for later metalogical work theorizing about the theory.

And third, PA's *non-logical axioms* – eliminating the redundancy we just noted from our original specification – are the following sentences:

Axiom 1. $\forall x(0 \neq Sx)$

Axiom 2. $\forall x\forall y(Sx = Sy \to x = y)$

Axiom 3. $\forall x(x + 0 = x)$

Axiom 4. $\forall x\forall y(x + Sy = S(x + y))$

Axiom 5. $\forall x(x \times 0 = 0)$

Axiom 6. $\forall x\forall y(x \times Sy = (x \times y) + x)$

plus (the closure of) every instance of the

Induction Schema $\{\varphi(0) \;\wedge\; \forall x(\varphi(x) \to \varphi(Sx))\} \;\to\; \forall x\varphi(x)$

where $\varphi(x)$ is an open wff of L_A that has 'x' (and perhaps other variables) free.

7.6 What can PA prove?

Even BA is good at proving quantifier-free equations. Q adds some ability to prove quantified wffs. We have so far noted just three additional simple quantified theorems that PA can prove. Further exploration reveals that a multitude of other familiar and not-so-familiar basic truths about the successor, addition, multiplication functions – and about the ordering relation (as defined in §6.9) – are provable in PA using induction.

In fact, we can prove so much that we might reasonably have hoped – at least before we'd heard of Gödel's incompleteness theorems – that PA would turn out to be a *complete* theory that indeed pins down all the truths of L_A.

Here is something else that would have encouraged this false hope, pre-Gödel. Suppose we define the language L_P to be L_A without the multiplication sign. Take P – so-called Presburger Arithmetic – to be the theory couched in the language L_P, whose axioms are Q's now familiar axioms for successor and addition, plus (the universal closures of) all instances of the induction schema that can be formed in the language L_P. In short, P is PA minus multiplication. *Then* P *is a negation-complete theory of successor and addition.* (We are not going to be able to prove that here – the argument uses a standard model-theoretic method called 'elimination of quantifiers' which isn't hard, and was known in the 1920s, but it would just take too long to explain.)

So the situation is as follows, and was known before Gödel proved his incompleteness theorem. (i) There is a complete formal axiomatized theory BA whose theorems are all the truths about successor, addition and multiplication expressible in the quantifier-free language L_B. (ii) There is another complete formal axiomatized theory P whose theorems are exactly the first-order truths expressible using just successor and addition. Against this background, the result that adding multiplication in order to get full PA gives us a theory which is incomplete and incompletable (if consistent) comes as a rather nasty surprise. It wasn't obviously predictable that adding multiplication would make all the difference. Yet it does. Indeed, as we've said before, as soon we have an effectively axiomatized arithmetic as strong as Q which has multiplication as well as addition, we get incompleteness.

And by the way, it isn't that a theory of multiplication must in itself be incomplete. In 1929, assuming we have a complete logic, Thoralf Skolem showed that there is a complete theory for the truths expressible in a suitable first-order language with multiplication but lacking addition or the successor function. So, when quantifiers are in play, why does putting multiplication together with addition and successor produce incompleteness? The answer will emerge shortly enough, but pivots on the fact that even a weak first-order arithmetic like Q with all three functions available can express/capture *all* 'primitive recursive' functions. But we'll have to wait until the next-but-one chapter to explain what *that* means.

7.7 A quick word about models of PA

I've spoilt the excitement! I've already said that despite its richness, PA is going to turn out to be an incomplete theory. We'll be able to find a Gödel sentence G_{PA} which is true on the intended interpretation built into the theory's language L_A, but such that PA can't prove it nor indeed disprove it.

PA's logic is complete – which means that the theory can formally prove every sentence which is semantically entailed by the axioms. Hence, since G_{PA} isn't provable, it can't be semantically entailed by the theory. Which means that there must also be some (non-standard) interpretation of the language which still makes all of PA's axioms true but which makes G_{PA} false. For the moment we are assuming that the standard interpretation built into L_A makes PA true: but now we see that, alongside its standard model, PA must have non-standard models.[6]

But actually, we don't need to appeal to Gödelian incompleteness to show *that*. For it quickly follows from two elementary theorems of model theory that PA (assuming all along that it is consistent) not only has non-standard models, but even has non-standard countable models – i.e. non-standard models whose domains are no bigger than the set of natural numbers.

The proof of this last claim is an optional extra. If the two theorems I'm about to state are unfamiliar, no matter – nothing in later chapters depends on following the short argument below. But it is fun, and worth knowing about.

So, the standard bits of model-theory we need are:[7]

Theorem 19. *(Downward) Löwenheim-Skolem. If the theory T (in a first-order language) has an infinite model at all, it has a countable model.*

Theorem 20. *Compactness. Let Σ be a set of sentences (in a first-order language): if every finite subset of Σ has a model, then so does Σ itself.*

And now, to get our argument going, suppose that we add to the language of PA a new constant c, and add to the axioms of PA the new axioms $\mathsf{c} \neq \mathsf{0}$, $\mathsf{c} \neq \overline{\mathsf{1}}$, $\mathsf{c} \neq \overline{\mathsf{2}}, \ldots, \mathsf{c} \neq \overline{\mathsf{n}}, \ldots$.

So we have added an infinite set of axioms which, taken all together, in effect tell us that there is a rogue object c which is not an eventual successor of zero. But still, each *finite* subset of the axioms of the expanded new theory has a model – just take the intended standard model of arithmetic and interpret c as denoting some number greater than the maximum n for which $\mathsf{c} \neq \overline{\mathsf{n}}$ is in the given finite subset of axioms.

Since each finite subset of the infinite set of axioms of the expanded theory has a model, the Compactness theorem tells us the *whole* expanded theory must have a model. This model evidently has to be infinite, so by the Löwenheim-

[6] Jargon reminder: we say an interpretation for a theory T (or for a set of sentences Σ) is a model of T (for Σ) iff it makes all T's theorems (all the sentences in Σ) true.

[7] These are easy corollaries of the usual completeness proof for first-order logic, and can be found in the standard textbooks.

Skolem theorem there will be in particular a countable model of this theory. This countable model will contain a 'zero', will have objects to be its eventual 'successors', but also have a rogue object c as the denotation of c (an object which isn't one of the eventual 'successors' of the 'zero'). So this model must be structured differently from the standard model of PA. However, since this countable structure is a model of PA-plus-some-extra-axioms it is, a fortiori, a model of PA. Hence PA has a non-standard countable model.[8]

Another elementary point of model theory tells us, however, that a theory can be negation-complete and yet still have multiple models that look different from each other. So the fact that PA has non-standard models doesn't rule out its being complete. So, to establish incompleteness, we do need to get down to our Gödelian arguments!

[8] "OK: that was smart! But can you now describe one of these countable-but-weird models of PA? In particular, what do the interpretations of the successor, addition and multiplication functions now look like?" The relevant functions take some effort to describe. For *Tennenbaum's Theorem* tells us that, for any non-standard model of PA, the interpretations of the addition and the multiplication functions can't be nice computable functions. But pursuing this further would take us too far off-piste.

8 Quantifier complexity

Wffs of the language L_A come in different degrees of *quantifier complexity*. We can distinguish, for a start, so-called Δ_0 , Σ_1, and Π_1 wffs. Later, in §**??**, we will find that the standard Gödel sentence that sort-of-says 'I am unprovable' is a Π_1 wff. This means that, while the Gödel sentence might be really long and messy, there is also a good sense in which it is logically quite simple. So what is a Π_1 wff? This short chapter explains.

8.1 Q knows about bounded quantifications

We often want to say that all/some numbers less than or equal to some bound have a certain property.

We can express such claims in formal arithmetics like Q and PA by using wffs of the shape $\forall x(x \leq \tau \rightarrow \varphi(x))$ and $\exists x(x \leq \tau \;\wedge\; \varphi(x))$. Here the term τ will specify the bound, $\varphi(x)$ expresses the relevant property, and $x \leq \tau$ is just short for $\exists v(v + x = \tau)$ (see §6.9: we assume τ doesn't contain v free). It is standard to further abbreviate such wffs by $(\forall x \leq \tau)\varphi(x)$ and $(\exists x \leq \tau)\varphi(x)$ respectively.

Now note that we have easy results like these:

1. For any n, $\mathsf{Q} \vdash \forall x(\{x = \overline{0} \vee x = \overline{1} \vee \ldots \vee x = \overline{n}\} \leftrightarrow x \leq \overline{n})$.
2. For any n, if $\mathsf{Q} \vdash \varphi(\overline{0}) \wedge \varphi(\overline{1}) \wedge \ldots \wedge \varphi(\overline{n})$, then $\mathsf{Q} \vdash (\forall x \leq \overline{n})\varphi(x)$.
3. For any n, if $\mathsf{Q} \vdash \varphi(\overline{0}) \vee \varphi(\overline{1}) \vee \ldots \vee \varphi(\overline{n})$, then $\mathsf{Q} \vdash (\exists x \leq \overline{n})\varphi(x)$.

Such results show that Q – and hence a stronger theory like PA – 'knows' that bounded universal quantifications with fixed numeral bounds behave like finite conjunctions, and that bounded existential quantifications with fixed numeral bounds behave like finite disjunctions. Later we will allow ourselves to appeal to simple results like these, without proof.

8.2 Δ_0 wffs

Let's now say that

Defn. 22. *An L_A wff is Δ_0 iff it can be built up from the non-logical vocabulary of L_A plus $\leq$ (defined as before), using the familiar propositional connectives, the identity sign, but only* bounded *quantifications.*

So, a Δ_0 wff is just like a quantifier-free L_A wff, except that (i) we are now allowed the existential quantifiers used in defining occurrences of $\leq$, and (ii) we can allow ourselves to wrap up some finite conjunctions into bounded universal quantifications, and similarly wrap up some finite disjunctions into bounded existential quantifications.

Given a Δ_0 wff is so like a quantifier-free wff, it should not come as a surprise to hear that we have the following result:

Theorem 21. *We can effectively decide the truth-value of any Δ_0 sentence.*

I won't give a full-dress proof. But, roughly speaking, we can unpack bounded quantifications into conjunctions or disjunctions (perhaps in a number of stages, if bounded quantifiers are nested one inside another). And then we are left with an equivalent wff built up using just propositional connectives from expressions of the form $\sigma = \tau$ and $\sigma \leq \tau$. But we can effectively compute the truth values of such basic expressions, and then use truth-tables to determine the values of wffs built up from them using the connnectives.

Since we can mechanically decide whether $\varphi(\overline{\mathsf{n}})$ is true given that the wff is Δ_0, this means that we can mechanically determine whether a Δ_0 open wff $\varphi(\mathsf{x})$ is satisfied by a given number n. In other words, a Δ_0 open wff $\varphi(\mathsf{x})$ will express a decidable property of numbers. Likewise a Δ_0 open wff $\varphi(\mathsf{x}, \mathsf{y})$ will express a decidable numerical relation. This will be important.

Now, since (i) Theorem 14 tells us that even Q can correctly decide all quantifier-free L_A sentences, (ii) Theorem 17 tells us that Q also knows about the relation $\leq$, and (iii) as just remarked in §8.1, Q knows that quantifications with fixed number bounds behave just like conjunctions/disjunctions, the next result won't be a surprise either:

Theorem 22. Q *(and hence* PA*) can correctly decide all Δ_0 sentences.*

Again, we won't spell out the details of the argument here (the simplest Δ_0 sentences are decidable, so we just need to show that building up more complex ones using connectives and bounded quantifiers preserves decidability).[1]

8.3 Σ_1 and Π_1 wffs

And now for the key definitions that we will need for future use:

Defn. 23. *An L_A wff is Σ_1 if it is (or is logically equivalent to) a Δ_0 wff preceded by zero, one, or more unbounded* existential *quantifiers. And a wff is Π_1 if it is (or is logically equivalent to) a Δ_0 wff preceded by zero, one, or more unbounded* universal *quantifiers.*

As a mnemonic, it is worth remarking that 'Σ' in the standard label 'Σ_1' comes from an old alternative symbol for the existential quantifier, as in $\Sigma x F x$ – that's a Greek 'S' for '(logical) sum'. Likewise the 'Π' in 'Π_1' comes from the corre-

[1] Enthusiasts can see the proof of Theorem 11.2 in *IGT2*.

sponding symbol for the universal quantifier, as in $\Pi x F x$ – that's a Greek 'P' for '(logical) product'.

Further, the subscript '1' in 'Σ_1' and 'Π_1' indicates that we are dealing with wffs which start with *one* block of similar quantifiers, respectively existential quantifiers and universal quantifiers.[2]

So a Σ_1 wff says that some number (pair of numbers, etc.) satisfies the decidable condition expressed by its Δ_0 core; likewise a Π_1 wff says that every number (pair of numbers, etc.) satisfies the decidable condition expressed by its Δ_0 core.

To check understanding, pause to make sure you understand why

1. The negation of a Δ_0 wff is still Δ_0;
2. A Δ_0 wff is also Σ_1 and Π_1;
3. The existential quantification of a Σ_1 wff is Σ_1; the universal quantification of a Π_1 wff is Π_1;
4. The negation of a Σ_1 wff is Π_1; the negation of a Π_1 wff is Σ_1.

(For (4), just recall the rules for exchanging the order of quantifiers and negations, and then use (1).)

Let's also note the following easy result:

Theorem 23. Q *can prove any true* Σ_1 *sentence (is '*Σ_1*-complete').*

Proof. Take, for example, a sentence of the type $\exists \mathsf{x} \exists \mathsf{y} \varphi(\mathsf{x}, \mathsf{y})$, where $\varphi(\mathsf{x}, \mathsf{y})$ is Δ_0. If the existentially quantified sentence is true, then for some pair of numbers m, n, the Δ_0 sentence $\varphi(\overline{\mathsf{m}}, \overline{\mathsf{n}})$ must be true. But by Theorem 22, Q proves $\varphi(\overline{\mathsf{m}}, \overline{\mathsf{n}})$; and hence it proves $\exists \mathsf{x} \exists \mathsf{y} \varphi(\mathsf{x}, \mathsf{y})$ by existential introduction.

Evidently the argument generalizes for any number of initial quantifiers, which shows that Q proves all truths which are (or are provably-in-Q equivalent to) some Δ_0 wff preceded by one or more unbounded existential quantifiers. ⊠

8.4 A remarkable corollary

Our last theorem looks entirely straightforward and unexciting, but it has an immediate corollary which is much more interesting:

Theorem 24. *If T is a consistent theory which includes* Q*, then every* Π_1 *sentence that it proves is true.*

Proof. Suppose T proves a *false* Π_1 sentence φ. Then $\neg\varphi$ will be a *true* Σ_1 sentence. But in that case, since T includes Q and so is 'Σ_1-complete', T will

[2]Just for the record, it is worth knowing that we can keep on going, to consider wffs with greater and greater quantifier complexity. So, we say a Π_2 wff is (or is logically equivalent to) one that starts with *two* blocks of quantifiers, a block of universal quantifiers followed by a block of existential quantifiers followed by a bounded kernel. Likewise, a Σ_2 wff is (equivalent to) one that starts with two blocks of quantifiers, a block of existential quantifiers followed by a block of universal quantifiers followed by a bounded kernel. And so it goes, up the so-called *arithmetical hierarchy* of increasing quantifier complexity. But for our purposes, we won't need to consider levels higher up the arithmetical hierarchy.

also prove $\neg\varphi$, making T inconsistent. Contraposing, if T is consistent, any Π_1 sentence it proves is true. ☒

Which is, in its way, a quite remarkable observation. It means that we don't have to fully *believe* a theory T – i.e. we don't have to accept that all its theorems are *true* on the interpretation built into T's language – in order to use it to establish that some Π_1 arithmetic generalization is true.

For example, with some minor trickery, we can state Fermat's Last Theorem as a Π_1 sentence. And famously, Andrew Wiles has shown how to derive this Π_1 sentence from some *extremely* heavy-duty infinitary mathematics. Now we see, intriguingly, that this background mathematical theory does not need to be *sound* (have true axioms) – whatever exactly that means when things get so very wildly infinitary! It is enough for Wiles's proof successfully to establish that Fermat's Last Theorem is true that his background theory is *consistent*. Remarkable!

8.5 Intermediate arithmetics

We said at the beginning of the previous chapter that, in moving on from the very weak arithmetics BA and Q to consider first-order PA, we were jumping over a whole family of theories of intermediate strength. We can now briefly describe those intermediate theories: they are the ones we get by restricting the quantifier complexity of suitable instances of the induction schema.

For example, $\mathsf{I\Sigma_1}$ is the theory we get by taking the first six axioms of PA (§7.5) plus the closure of every instance of the Induction Schema

$$\{\varphi(0) \land \forall x(\varphi(x) \to \varphi(Sx))\} \to \forall x\varphi(x),$$

where $\varphi(x)$ is now an open Σ_1 wff of L_A.

There is *technical* interest in knowing how much a theory like $\mathsf{I\Sigma_1}$ can prove (as we will see in §**??**). But do such theories have any *conceptual* interest? After all, we gave reasons in §7.3 for being generous with induction: we asked, if $\varphi(x)$ expresses a genuine arithmetical property, how can induction fail for $\varphi(x)$?

But let's backtrack for a moment and ask again: which L_A open wffs $\varphi(x)$ *do* express genuine properties? Previously we took it that they all do (even if, in the general case, we may not be able to decide whether a given number n has the property or not): that is why we said that any such wff $\varphi(x)$ is 'suitable' for appearing in an instance of the Induction Schema. But suppose you are a *very* stern constructivist who thinks that an expression $\varphi(x)$ only really, *really*, makes sense if it is Δ_0 and so we can effectively decide whether or not it holds true of a given number. Or perhaps, slightly less sternly, you allow that the expression makes sense if it is Σ_1 so at least we can prove it true of a given number when it is. *Then* you can reasonably want to restrict the induction principle to suitable instances using only Δ_0 (or Σ_1) expressions. But it would take us far too far afield even to begin to explore the merits of such stern proposals here.

Interlude

Let's pause to draw breath and take stock.

1. In Chapter 2 we met the First Incompleteness Theorem in this rough form: a nice enough theory T (which contains the language of basic arithmetic) will always be negation incomplete – there will always be sentences of basic arithmetic it can neither prove nor disprove.

2. We then noted in Chapter 3 that we can cash out the idea of being a 'nice enough' theory in two ways. We can assume T to be *sound*. Or, retreating from that semantic assumption, we can require T to be a *consistent theory which proves a modest amount of arithmetic*. Gödel himself highlights the second version.

3. Of course, we didn't *prove* the Theorem in either version, there at the very outset. However, in Chapter 4, we waved an arm rather airily at the basic strategy that Gödel uses to establish the Theorem – namely we 'arithmetize syntax' (i.e. numerically code up facts about provability in ways that we can express in formal arithmetic) and then construct a Gödel sentence for a theory T that is true if and only if is isn't provable-in-T.

4. In Chapter 5 we did a bit better, in the sense that we actually gave a *proof* that a consistent, effectively axiomatized, sufficiently strong, formal theory cannot be negation complete.

   The argument was revealing, because it shows that we can get incompleteness results without calling on the arithmetization of syntax and the construction of Gödel sentences. However, the argument depends on the notion of 'sufficient strength' which is defined in terms of the informal notion of a 'decidable property' (a theory, remember, is sufficiently strong if it captures every decidable property of the natural numbers). And the discussion in Chapter 5 doesn't explain how we can sharpen up that informal notion of a decidable property, nor does it explain what a sufficiently strong theory might look like.

5. We need to get less abstract, and start thinking about specific theories of arithmetic. In Chapter 6, as a warm-up exercise, we first looked at BA, the quantifier-free arithmetic of the addition and multiplication of particular numbers. This is a negation-complete and decidable theory – but of

course the theory is only complete, i.e. is only able to decide every sentence constructible in its language, because its quantifier-free language is so very limited. However, if we augment the language of BA by allowing ourselves the usual apparatus of first-order quantification, and replace the schematically presented axioms of BA with their obvious universally quantified correlates (and add in the axiom that every number bar zero is a successor) we get the much more interesting Robinson Arithmetic Q.

Since we are considerably enriching what can be expressed in our arithmetic language while not greatly increasing the power of our axioms, it is no surprise that Q is negation incomplete. And we can prove this without any fancy Gödelian considerations. We can easily show, for example, that Q can't prove either $\forall x(0 + x = x)$ or its negation. Q, then, is a very weak arithmetic. Still, it will turn out to be the 'modest amount of arithmetic' needed to get a syntactic version of the First Theorem to fly. We announced (but of course haven't proved) that even Q is sufficiently strong: which explains why Q turns out to be so interesting despite its weakness.

6. In Chapter 7, we then moved on to introduce first-order Peano Arithmetic PA, which adds to Q a whole suite of induction axioms (every instance of the Induction Schema). Exploration reveals that this theory, in contrast to Q, is very rich and powerful. We might, pre-Gödel, have very reasonably supposed that it is a negation-complete theory of the arithmetic of addition and multiplication. But the theory is still effectively axiomatized, and the First Theorem is going to apply (assuming PA is sound, or is at least consistent and satisfies another syntactic condition). So PA too will turn out to be negation incomplete.

7. There are theories intermediate in strength between Q and PA, theories which have induction axioms but only for wffs up to some degree of quantificational complexity. For technical reasons, we will later be interested in one such intermediate theory (Chapter **??**). But the task of Chapter 8 was just to explain this notion of quantificational complexity, and in particular explain what Σ_1 and Π_1 wffs are.

Which brings us up to the current point in this book.

To give a sense of direction, let's next outline where we are going in the next five chapters. (Skip if you don't want spoilers!)

8. The formal theories of arithmetic that we've looked at so far have (at most) the successor function, addition and multiplication built in. But why stop there? Even high-school arithmetic acknowledges many more numerical functions, like the factorial and the exponential.

   Chapter **??** describes a very wide class of such numerical functions, the so-called primitive recursive (p.r.) ones. They are a major subclass of the effectively computable functions.

   We also define the primitive recursive properties and relations – a numerical property/relation is p.r. when some p.r. function can effectively decide when it holds.

9. Chapter **??** then shows that L_A, the now-familiar formal language of basic arithmetic, can *express* all p.r. functions and relations.

   Moreover Q and hence PA can *capture* all those functions and relations too – i.e. they can case-by-case prove wffs that assign the right values to the functions for particular numerical arguments. So Q and PA, despite having only successor, addition and multiplication 'built in', can actually deal with a vast range of functions (at least in so far as they can 'calculate' the value of the functions for arbitrary numerical inputs).

   Note the link with our earlier talk about 'sufficiently strong theories' (Defn. 18). Those, recall, are theories that can capture all effectively decidable properties of numbers. Well, now we are going to show that PA (indeed, even Q) can capture at least all those effectively decidable properties of numbers which are primitive recursive. And we'll find that that's enough for the core Gödelian argument to go through.

10. In Chapter **??** we then introduce again the key idea of the 'arithmetization of syntax' by Gödel-numbering, an idea which we first met in §§4.3 and 4.4. Focus on PA for the moment, and fix on a suitable Gödel-numbering. Then we can define various numerical properties/relations such as:

    Wff(n) iff n is the code number of a PA-wff;
    Sent(n) iff n is the code number of a PA-sentence;
    Prf(m, n) iff m is the code number of a PA-proof of the sentence with code number n.

    Moreover – the crucial result – these properties/relations are primitive recursive. Similar results obtain for any sensibly axiomatized formal theory.

11. Since *Prf* is p.r., and the theory PA can express all p.r. relations, we can express some facts about PA-proofs in PA itself. In chapter **??** we use this fact in constructing a Gödel sentence which is true if and only if it is not provable in PA. We can thereby prove a semantic version of Gödel's first incompleteness theorem for PA in something close to Gödel's way, assuming PA is sound. The result generalizes to other sensibly axiomatized sound arithmetics that include Q.

12. Then Chapter **??**, at last, proves a crucial syntactic version of the First Incompleteness Theorem, again in something close to Gödel's way.

So now read on . . . !