Preface

Why these lecture notes? After all, I have already written a rather long book, *An Introduction to Gödel's Theorems* (originally 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, the 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 to someone prepared to put in the required time and effort, there is – to be frank – a *lot* more in the book than is really needed by those wanting a first encounter with the famous incompleteness theorems.

Some readers might therefore appreciate a cut-down version of some of the material in the book – an introduction to the *Introduction*, if you like. Hence *Gödel Without (Too Many) Tears*. There are occasional references here to sections of the book, pointing to 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 these notes.

A first version – call it *GWT1* – was written to accompany the last few outings of a short lecture course given in Cambridge (which was also repeated at the University of Canterbury, NZ). Many thanks to many students for useful feedback.

GWT1 was intended to bridge the gap between classroom talk'n'chalk which just highlighted the Really Big Ideas, and the more detailed treatments of topics now available in my book. However, despite that intended role, I did try to make *GWT1* reasonably stand-alone.

Those notes were tied to the first edition of my book, *IGT1*, as published in 2007. A significantly improved second edition of the book, *IGT2*, was published in 2013. So I updated *GWT1* in 2014 to make a corresponding second version of the notes – call it *GWT2*.

It's time to revisit the notes, and make some minor improvements. So here is *GWT3*.

Who are these notes for? Someone who wants more than an arm-waving informal discussion, who wants to understand what Gödel's incompleteness theorems say and have some real sense of how they can be proved. There isn't a lot of purely philosophical discussion here: the aim, rather, is to put you in a position where you have a secure enough initial understanding of what's going on logically

that you can then sensibly make a start on thinking about the philosophical implications.

What background in logic do we presuppose? What do you need to bring to the party? Very little. If you have done a standard introductory logic course, and have the patience to follow some simple mathematical arguments, you should have little difficulty in following the exposition here.

On notation: you probably don't need to be told but, just in case, 'iff' abbreviates 'if and only if', and '□' marks the end of a proof.

A number of people kindly let me know about typos and more serious mistakes in earlier editions: I should in particular mention Henning Makholm. I have no doubt introduced some more mistakes in this new edition. So please let me know by emailing peter_smith@logicmatters.net.

1 Incompleteness, the very idea

1.1 A 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 1938, 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 first-order predicate calculus (showing that a standard proof system for first-order logic indeed captures 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 on models of General Relativity with 'closed timelike curves' (where, in some sense, time travel is possible). Always a perfectionist, he became a very reluctant publisher (some of his philosophically most interesting work is in the volume of Unpublished Essays and Lectures in his *Collected Works*).

Talk of 'Gödel's Theorems' typically refers, however, to the two *incompleteness* theorems presented in an epoch-making 1931 paper. And it is these theorems, and more particularly, the First Theorem, that these notes are all about. (Yes, that's right: Gödel proved a 'completeness theorem' and also 'incompleteness theorems'. We'll explain the difference in a moment!)

The impact of the incompleteness theorems on foundational studies is hard to exaggerate. For, putting it crudely and a little bit tendentiously, they sabotage the ambitions of two major foundational programs – logicism and Hilbert's programme. We'll say just a little about logicism in this chapter, and something about Hilbert's programme much later, in Chapter 15, when we get round to discussing the Second Theorem. But you don't have to know anything about this background to find the two theorems intrinsically fascinating.

1.2 The idea of an axiomatized formal theory

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 it was intended to be the first part of what was

going to be a two part paper, with Part II spelling out 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 Gödel thought he didn't need to bother.

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, now more than a century old and 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. A 'system' (in the relevant sense) is an axiomatized theory – or more precisely, an *effectively axiomatized formal theory*. But what does that mean?

The general idea of an axiomatized formal theory is no doubt familiar. Roughly: you fix on a formalized language, set down some axioms stated in that language, specify some apparatus for formally deriving results from your axioms, and there you have a theory. But now we need to be more explicit: 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,
(iii) an effectively formalized proof system.

We'll explain these headlines in just a moment. But first, 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 there's an algorithm (a finite set of instructions for a deterministic computation) for settling in a finite number of steps, for any object $o \in D$, whether o has property P.*

Likewise, a set Σ is effectively decidable iff the property of being a member of that set is effectively decidable.

Relatedly, the answer to a question Q is effectively determinable iff there is an algorithm which gives the answer (again by a deterministic computation, in a finite number of steps).

To put it another way, a property is effectively 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 do the trick (idealizing away from practical constraints of time, etc.).

Here are a couple of familiar examples from elementary logic: the property of being a tautology is effectively decidable (by a truth-table test!); and we can effectively determine what the main connective of a sentence is (by some bracket counting!).

How satisfactory are our definitions, though? We've just invoked the idea of what a computer (in principle) could do by implementing some algorithm. But

doesn't that leave quite a bit of slack in the definition? Why shouldn't what a computer can do depend, for example, on its architecture (even given that we are idealizing, and e.g. putting no time limit on its computations, or the amount of memory-space needed)?

It turns out that the notion of effective decidability is in fact very robust: what is algorithmically-computable-in-principle according to one sensible sharpened-up definition turns out to be exactly what is algorithmically-computable-in-principle according to any other sensible sharpened-up definition. Of course, it's not at all trivial 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* be put into good shape by sharpening the notion of effective decidability.

Against this background we can now explain those 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 familiar from earlier logic courses. But note, 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 unique 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 – in my idiolect – that's redundant.

The familiar way of presenting the syntax of a formal language is by (a) specifying some finite set of symbols,[1] and then giving rules for building up expressions from these symbols. And we standardly do this in such a way that (b) we can *effectively* decide whether a given string of symbols counts as a term or wff or a wff with one free variable or a sentence (we can give algorithms which decide well-formedness, etc.).

The familiar way of presenting the semantics is then to assign semantic values to the non-logical expressions of the language, fix domains of quantification, and then give rules for working out the truth-conditions of longer and longer expressions in terms of the way they are syntactically built up from their parts. In a standard formal language, we can *effectively* recover from a sentence its 'constructional history', i.e. mechanically determine the way it is syntactically built up from its parts; then, relying on this information, (c) we can use the semantic rules to mechanically work out the interpretation of any given sentence.

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

(Read that carefully! What we can mechanically work out is what the sentence *says*. But it is 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*!)

Let's say that a formalized language which shares these characteristics (a), (b) and (c) is effectively formalized. So, in sum,

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 effectively decidable and the syntactic structure of any sentence is effectively determinable, and (c) this syntactic structure together with the semantic rules can be used to effectively determine the unique intended interpretation of every sentence.*

Why do we want (b) the syntactic properties of being a sentence, etc., to be effectively decidable? Well, the point of setting up a formal language is, for a start, to put issues of what is and isn't e.g. a 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. 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) Some logic books define a theory to be just any old set of sentences. We are concerned, though, with the narrower notion of an *axiomatized theory*. We highlight some bunch of sentences Σ as giving *axioms* for the theory T; we 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 theory, apart from its using an effectively formalized language? For a start, we require it to be effectively decidable what's an axiom of T. 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 is indeed 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 deductive proof system, a set of rules for deducing further theorems from our initial axioms. But a proof system such that we couldn't routinely tell whether its rules are in fact being followed again wouldn't have much point for practical purposes. Hence we naturally also require that a theory has an effectively formalized proof system, i.e. one where it is effectively decidable whether a given array of wffs is indeed a well-constructed proof from the axioms according to the rules of the deductive system.

Note, it doesn't matter for our purposes whether the proof system is e.g. a Frege/Hilbert axiomatic logic, a natural deduction system, a tree/tableau system, or a sequent calculus – so long as it is indeed effectively checkable that a candidate proof-array has the property of being properly constructed according to the rules of the proof system.

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 indeed a proof from the axioms according to the rules.*

Careful, though! To say that, for a properly formalized theory T it must be effectively decidable whether a candidate T-proof of φ is indeed a kosher proof is not, repeat *not*, to say that it must be effectively decidable whether φ actually *has* a 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 4 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.)

1.3 'Formally undecidable propositions' and negation incompleteness

Henceforth, when we talk about theories, we always mean effectively axiomatized formal theories (unless we explicitly say otherwise).

Some familiar logical notation, applied to formal theories:

Defn. 4. *'$T \vdash \varphi$' says: there is a formal deduction in T's proof system from T-axioms to the sentence φ as conclusion (in short, φ is a T-theorem).*

Defn. 5. *'$T \models \varphi$' says: any model (re)interpreting the non-logical vocabulary that makes all the axioms of T true makes φ true.*

So '$\vdash$' officially signifies *provability* in T, which is a syntactically-definable relation. While '$\models$' signifies *logical entailment*, a semantic relation defined by generalizing over interpretations.[2]

[2] 'You said that a theory's language has a built-in interpretation: now you seem to be forgetting that.' Not so. Recall, some premisses *logically* entail a given conclusion if the inference from the premisses to the conclusion is necessarily truth-preserving *just in virtue of the distribution of logical operators in the relevant sentences*. That's why our official definition of the notion abstracts away from the given meaning of the non-logical constituents of the sentences while keeping the meaning of the logical apparatus fixed, and generalizes over all possible reinterpretations of the non-logical constituents.

Of course, we normally want a formal deduction to indeed be genuinely truth-preserving; so we will want our proof system to respect logical entailments, requiring that $T \vdash \varphi$ only if $T \vDash \varphi$. In a word, we require a deductive system in a sensible theory to be *sound*.

We can't in general insist on the converse, however. But take the important special case where the theory T has a standard first-order logical system. In a classical first-order setting, if an inference from T to φ is semantically valid, i.e. is necessarily truth-preserving, then there will be a formal deduction of φ from the axioms of T. This was shown for a Hilbert-style deductive system by Gödel in his doctoral thesis: hence *Gödel's completeness theorem*.

Some more key definitions. We will be interested in what claims a theory T can settle, one way or the other. So 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,

Defn. 7. *A sentence φ is* formally undecidable *by T iff $T \nvdash \varphi$ and $T \nvdash \neg\varphi$.*

A related bit of terminology:

Defn. 8. *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 two notions of completeness, if we take a toy example. Suppose theory T is built in a propositional language with just three propositional atoms, $\mathsf{p}, \mathsf{q}, \mathsf{r}$, and the usual propositional connectives. We give T a standard propositional classical logic (pick your favourite flavour of system!). And assign T just a single non-logical axiom: $(\mathsf{p} \wedge \neg\mathsf{r})$.

Then, by assumption, T has a *semantically-complete logic*, since standard propositional calculi are complete. That is to say, 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 basic assumptions about the matter in hand!

But 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. And then we'd ideally like to lay down enough axioms framed in L such that, for any L-sentence φ, then

$T \vdash \varphi$ just when φ is true. So, making the classical assumption that for any sentence φ, either φ is true or $\neg\varphi$ is true, we'd very much like T to be such that either $T \vdash \varphi$ or $T \vdash \neg\varphi$ (but not both!).

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

1.4 Seeking a negation-complete theory of arithmetic

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

Let's first fix on a formal *language of basic arithmetic* in which we can regiment elementary arithmetical propositions. We will give this language

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

We will also give this language

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

(We aren't building in subtraction and division as primitives, however. 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 off for ever: moreover there are no strays – i.e. every natural number is in that sequence starting from zero. It is surely plausible to suppose that (a) and (b) together should indeed 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 characterize (a) the number series, and (b) 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) theory of basic arithmetic.

1.5 Logicism and *Principia*

It is natural to ask: what could be the *status* of the axioms of a formal theory of arithmetic – e.g. the status 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 the 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 are *analytic*, simply truths of logic-plus-definitions. On this view, we don't need Kantian 'intuitions' going beyond logic: logical reasoning from definitions is enough to get us the axioms of arithmetic, and more logic gives us the rest of the arithmetic truths from these axioms. This Fregean line is standardly dubbed *logicism*.

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

That 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 in 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?

1.6 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. And there are technical complications which mean that 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 states that there is an infinite number of objects; and then there is the notoriously dodgy so-called *Axiom of Reducibility*. But we don't need to go into details; for we can leave those worries aside – they pale into insignificance compared with the bombshell exploded by Gödel.

For Gödel's First Incompleteness Theorem sabotages not just the grand project of *Principia*, but shows that *any* 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 very rough first shot – that *nice theories containing enough arithmetic are always negation incomplete*: for any nice theory T, there will be arithmetic truths that can't be proved in that particular theory.

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![3]

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 3 we outline a Gödelian proof of one version.

[3] '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, while allowing that we can't capture the relevant entailment relation in a single properly axiomatized deductive system of logic. Then the resulting overall system of arithmetic won't count as a formal axiomatized theory of all arithmetical truth since its proof system is not effectively formalizable, and Gödel's theorems don't apply. But all that is another story.

2 The First Theorem, two versions

2.1 Soundness, consistency, etc.

Let's read into the record two standard definitions:

Defn. 9. *A theory T is* sound *iff its axioms are true (on the interpretation built into T's language), and its logic is truth-preserving, so all its theorems are true.*

Defn. 10. *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 $T \vdash \psi$ for all ψ. And of course, soundness implies consistency. We shouldn't need to delay over these no doubt familiar ideas.

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

Defn. 11. *The formalized interpreted language L* contains the language of basic arithmetic *if 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.

(OK, you might worry whether the natural number system referred to in set theory is the genuine article or just a structurally equivalent surrogate. But then what is 'the genuine article'? We are not going to tangle with *that* messy issue, as we have quite enough other things to worry about! When we talk of a theory quantifying over numbers, then, take it to be quantifying over numbers or over whatever surrogates we can take to play the role of natural numbers. Nothing relevant to our project hangs on the difference.)

2.2 Two theorems distinguished

In his 1931 paper, Gödel proves (or rather more accurately, gives us most of the materials to prove) the following:

Theorem 1. *Suppose T is a formal axiomatized 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 a very gappy way!) 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 the 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 think that we don't have to 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 you don't need to deploy any semantic notions to get an incompleteness result. So he demonstrates (a version of) the following:

Theorem 2. *Suppose T is a formal axiomatized 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 a certain 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 (in the relevant sense) 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 in due course; but this indicates the general *character* of Gödel's central result. 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 to spell out just how much arithmetic that is, though we'll eventually find that it is stunningly little. 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 condition again.

For now, then, the first take-away message of this chapter is that the incompleteness theorem does come in two flavours. There's a version making a *semantic* assumption (the relevant theory T needs to expressively rich enough

and sound), and there's a version making only a *syntactic* assumption (about what T can derive from its axioms). It is important to keep this firmly in mind.

2.3 Incompleteness and incompletability

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

Suppose 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*, say that G_T is 'absolutely unprovable', whatever that could possibly mean. It just says that G_T and its negation are unprovable-in-T.

Ok, you might well ask, why don't we just '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, the added new axiom is true, and its 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 still remains negation-incomplete, unless it stops being effectively axiomatized. So here's the second important 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*.[1] (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.

2.4 The completeness and incompleteness theorems

A reality check. We've already made the distinction we need in §1.3, and we illustrated it then with a toy example. But experience suggests that it will do no harm at all to repeat the point!

A semantic completeness theorem of the kind you are no doubt familiar with from elementary logic is about the relation between semantic and syntactic consequence relations. For example, you will know about this completeness result:

> If T is a theory cast in a first-order language with a standard first-order deductive apparatus, then for any φ, if $T \vDash \varphi$ then $T \vdash \varphi$.

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

(That's Gödel's completeness theorem.) But it doesn't follow from T's having a *complete logic* like this that T is a negation-complete *theory*. For example, if T is a theory of arithmetic with a standard first-order deductive apparatus it certainly has a semantically complete logic; but it can easily be a negation-incomplete theory. 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 $T \vdash \neg\varphi$!

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

3 Outlining a Gödelian proof

3.1 A notational convention

Before continuing, we should highlight a very useful notational convention that we have already been using and which we will continue to use throughout these notes:

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

3. Greek letters, like 'Σ' and 'φ', are schematic variables in the metalanguage, which we can use e.g. in generalizing about wffs of our formal systems.

The same convention is used in *IGT2*, and versions of it are quite common elsewhere too. There's a lot of to-and-fro in this book between claims of informal mathematics, samples of formal expressions and formal proofs, and general claims about formal proofs. It is essential to be clear which is which, and our notational convention should help considerably.

3.2 Formally expressing numerical properties, relations and functions

In the next few sections, we are going to prepare the ground for an outline sketch of how Gödel proved (a version of) Theorem 1.

We start with a couple more definitions. Recall, we said that the standard numerals of a language of basic arithmetic are the expressions '$\mathsf{0}$', '$\mathsf{S0}$', '$\mathsf{SS0}$', '$\mathsf{SSS0}$', ... Let's now introduce a handy notational convention:

Defn. 12. *We will use '$\overline{\mathsf{n}}$' to abbreviate the numeral denoting the number n.*

So '$\overline{\mathsf{n}}$' will consist of n occurrences of 'S' followed by '$\mathsf{0}$'.

Now assume we are dealing with a language L which has standard numerals (and for the moment we'll also assume L has the usual apparatus of variables). Then we will say:

Defn. 13. *The open wff $\varphi(\mathsf{x})$ of the language L* expresses *the numerical property P iff $\varphi(\overline{\mathsf{n}})$ is true on interpretation just when n has property P. Similarly, the formal wff $\psi(\mathsf{x},\mathsf{y})$ expresses the numerical two-place relation R iff $\psi(\overline{\mathsf{m}},\overline{\mathsf{n}})$ is true just when m has relation R to n. And the formal wff $\chi(\mathsf{x},\mathsf{y})$ expresses the numerical one-argument function f iff $\chi(\overline{\mathsf{m}},\overline{\mathsf{n}})$ is true just when $f(m) = n$.*

Hopefully, this definition should seem entirely natural.[1] For example, 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. 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. More generally: expressing is just a matter of having the right extension.

Though we won't need it, the generalization of our definition to cover expressing many-place relations and many-argument functions is obvious enough.

3.3 Gödel numbers

And now for a key 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 perhaps not in such everyday currency in 1931. But even then, the following sort of definition should have looked quite unproblematic:

Defn. 14. *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. There is an algorithm for sending an expression (or sequence of expressions) to a number; and an algorithm for undoing the coding, sending a code number back to the expression (sequence of expressions) 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 seven basic symbols. Associate those with the digits 1 to 7, and associate the comma we might use to separate expressions in a sequence of expressions with the digit 8. 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 mechanical (though if the string of digits expressing some number contains '9' or '0', the algorithm won't output any result when we try to decode it).

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

'$\varphi(\overline{\mathsf{n}})$' indicates, of course, the result of replacing the variable 'x' in '$\varphi(\mathsf{x})$' by the standard numeral for n. But you knew that!

Which scheme of Gödel-numbering we adopt in practice will depend on considerations of ease of manipulation. In theory it won't matter: 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).

3.4 Three new numerical properties/relations

Defn. 15. *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,*

> *Wff*(n) *iff* n *is the Gödel number of a* T*-wff.*
> *Sent*(n) *iff* n *is the Gödel number of a* T*-sentence.*
> *Prf*(m, n) *iff* m *is the Gödel number of a* T*-proof of the* T*-sentence with code number* n*.*

Now, true enough, these 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 properties/relations* Wff, Sent, Prf *are effectively decidable.*

Proof. Take *Wff*. The number n has this property if and only if (i) n decodes into a string of T-symbols (an effective process a computer could carry out), and (ii) that string of T-symbols is a wff (which, since T has an effectively formalized language by assumption, again a computer could decide). In short, it is effectively decidable whether *Wff*(n).

The case of *Sent* is similar. And as for *Prf*, 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 indeed a T-proof of the conclusion coded by n. That is to say, it is effectively decidable whether *Prf*(m, n). □

3.5 T can express *Prf*

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 Gödelian proof of Theorem 1. For the moment, we will have to state the results without detailed proof: later, we will see what it takes to prove (close variants) of them. But for now, we just want to explain what the two results claim. The first is as follows:

Theorem 4. *Suppose T is an effectively axiomatized 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 property Prf using some arithmetical wff* $\mathsf{Prf}(\mathsf{x}, \mathsf{y})$.

This is not supposed to be obvious! It takes quite a bit of effort to show how to build – just out of the materials of the language of basic arithmetic – a formal T-wff we'll abbreviate $\mathsf{Prf}(\mathsf{x}, \mathsf{y})$ that expresses the property *Prf*, so $\mathsf{Prf}(\overline{\mathsf{m}}, \overline{\mathsf{n}})$ is true exactly when $Prf(m, n)$, i.e. when m is the code number of a T-proof of the wff with number n.

How do we show this surprising claim? As I said, we are not going to spell this out right now. But, to a first approximation, we can rely on the fact that the language of basic arithmetic turns out to be *really* good at expressing decidable numerical properties and relations, and we've just seen that the numerical relation *Prf* is decidable because T is a formalized theory.

Or rather, to a better approximation, we rely on the fact that basic arithmetic is very good at expressing so-called *primitive recursive* relations and for any sensible theories $Prf(m, n)$ is primitive recursive. The idea of a primitive recursive relation is a simple but technically defined notion that covers a large class of intuitively effectively decidable relations. So, for our purposes, we can trade in the informal notion of a decidable relation for the crisply defined notion of a primitive recursive relation. More about this later (in Chapter 10).

3.6 Defining a Gödel sentence G_T

It's useful to start adding subscripts to emphasize which theory we are dealing with. With a predicate $\mathsf{Prf}_T(\mathsf{x}, \mathsf{y})$ available in the theory T to express the relation Prf_T, we can now add a further neat definition:

Defn. 16. *Put* $\mathsf{Prov}_T(\mathsf{y}) =_{\text{def}} \exists \mathsf{x} \mathsf{Prf}_T(\mathsf{x}, \mathsf{y})$ *(where the quantifier runs over all the numbers in the domain). Then* $\mathsf{Prov}_T(\overline{\mathsf{n}})$ *says that some number Gödel-numbers a T-proof of the wff with Gödel-number n, i.e. the wff with code number n is a T-theorem. So* $\mathsf{Prov}_T(\mathsf{x})$ *is naturally called a* provability predicate.

And now comes another key result we need for building towards the First Theorem. Still working with the same theory T and Gödel-numbering scheme,

Theorem 5. *We can construct a* Gödel sentence G_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.

This construction involves a clever but surprisingly easy trick: we won't spell it out now, so again we'll delay the proof of this theorem. For the moment, just note what our theorem implies: by construction, 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 theorem, i.e. iff G_T is not a theorem. In short, G_T *is true if and only if it isn't a theorem.*

Stretching a point, it is rather as if G_T 'says' *I am unprovable in* T. (But that *is* stretching a point: 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 any wff. More about this later, in §11.3.) Still, with that point in mind, you'll probably immediately spot that we can now prove ...

3.7 Incompleteness!

Theorem 1. *Suppose T is a formal axiomatized 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 $T \vdash \mathsf{G}_T$. Then G_T would be a theorem, and hence G_T – which is true iff it is not provable – 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 – which is true iff it is not provable – 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 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.

For future reference, Gödel's proof of the syntactic version of the incompleteness theorem, i.e. Theorem 2, uses the same construction of a Gödel sentence, but this time we need to trade in the semantic assumption that T is sound for the syntactic assumption that T is consistent and can prove some basic arithmetical truths (and we require T to have that currently mysterious 'additional desirable [syntactic] property'). So we will need syntactic analogues of Theorems 4 and 5. Again more devilish detail. Again more about this in due course.

3.8 Gödel and the Liar

Of course, you might well think that there is something a bit worrying about our sketch in the last section. For basically, I'm saying we can construct an arithmetic sentence G_T in T that, via the Gödel number coding, is true if and only if to 'G_T' *is not provable in* T, and then such a sentence can neither be proved nor refuted in a sound T. But shouldn't we be suspicious about this idea? After all, we know we fall into paradox if we try to construct a Liar sentence L which holds if and only if 'L' *is not true*. So why does the self-reference in the Liar sentence lead to *paradox*, while the self-reference in Gödel's proof give us a *theorem*?

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

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 §14.4.

4 Undecidability and incompleteness

In Chapter 1, we introduced the very idea of a negation-incomplete, effectively axiomatized, formal theory T.

We noted that if we are aiming to construct a theory of basic arithmetic, we would ideally like the theory to be able to prove *all* the truths expressible in the language of basic arithmetic, and hence to be negation complete (at least as far as statements of basic arithmetic are concerned). But Gödel's First Incompleteness Theorem tells us that that's impossible: roughly, a nice enough theory T will always be negation incomplete for basic arithmetic.

Now, as we noted in Chapter 2, 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 idea of T's being a *consistent theory which proves enough arithmetic*. Then we saw in Chapter 3 that Gödel's own proofs, of either flavour, go via the idea of numerically coding up inside arithmetic itself syntactic facts about what can be proved in T, and then constructing an arithmetical sentence that – via 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).

We ended by noting that, at least at the level of arm-waving description of Chapter 3, the Gödelian construction might look a bit worrying. After all, we all know that self-reference is dangerous – think Liar Paradox! So is Gödel's construction entirely legitimate?

It certainly is, as should become quite clear over the coming chapters. But I think it might well go a little way towards calming the worry that some illegitimate trick is being pulled, and it is certainly of intrinsic interest, if we first give a somewhat different sort of proof of incompleteness, one that doesn't go via any explicitly self-referential construction. This proof will, however, introduce the idea of a *diagonalization argument*. And as we will see later, it is in fact 'diagonalization' rather than self-reference which is really the key to Gödel's own proof.

So now read on ...

4.1 Negation completeness and decidability

Let's start with another definition:

Defn. 17. *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 $T \vdash \varphi$.*

(Terminology check: a theory T formally *decides* a sentence φ iff either $T \vdash \varphi$ or $T \vdash \neg\varphi$; a theory T is *decidable* iff for any φ we can effectively determine whether $T \vdash \varphi$. Two different notions then with similar terminology: in practice, though, you shouldn't get confused![1])

We can now easily show:

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 sequences of symbols, it will be a mechanical matter to decide whether a given string is in fact a sequence of 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 indeed a kosher well-constructed proof, finishing with a sentence, then list this last wff φ as a T-theorem.

We can in this way start mechanically generating a list that will eventually contain any T-theorem (since any T-theorem is the last sentence in 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. □

[1] To fix ideas, note that a theory can be decidable without deciding every wff. For example, the toy propositional theory T of §1.3 is decidable (as is familiar, because propositional logic is complete, a truth-table test will determine whether $T \vdash \varphi$ for any given wff φ of T's language). In particular, we can see that $T \nvdash \mathsf{q}$ and $T \nvdash \neg\mathsf{q}$. Therefore T doesn't decide q, so T doesn't decide every wff.

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 procedure in our official generous sense (see the comments again on Defn. 1, or the further explanation in *IGT2*, §3.1).

4.2 Capturing numerical properties in a theory

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

Defn. 13. *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.

(Recall, $\overline{\mathsf{n}}$ indicates L's standard numeral for n.) And now we want a new companion definition:

Defn. 18. *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; what a theory can *capture* – mnemonic: case-by-case prove – depends on the richness of its axioms and rules of inferences. (To be honest, 'represents' is much more commonly used than my 'captures', but I'll stick here to the slightly idiosyncratic but memorable jargon adopted in *IGT2*.)

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 18. *(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'll want any theory that aims to deal with arithmetic not just to express but to capture lots of numerical properties, i.e. to prove which particular numbers have or lack these properties. But what particular sort 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 don't want going formal to *diminish* our ability to determine whether n has a 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 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 to capture P in the sense of our definition.

The suggestion therefore is that, if P is any effectively decidable property of numbers, we ideally want a competent theory of arithmetic T to be able to capture P. Which motivates the following definition:

Defn. 19. *A formal theory T is sufficiently strong iff it captures all decidable numerical properties.*

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

In sum: it seems a reasonable and desirable condition on an ideal formal theory of the arithmetic of the natural numbers that it be sufficiently strong – when *we* can (or at least, given world enough and time, *could*) decide whether a particular number has a certain property, the *theory* can do it.

4.3 Sufficiently strong theories are undecidable

We now prove a lovely theorem (take it 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 (for expressing numerical properties). And by Defn 2, it must in fact be decidable what strings of symbols are T-wffs with the free variable 'x'. And we can use the dodge 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 churn out all the strings of symbols of T's language 'in alphabetical order', and then mechanically select out the wffs with free variable 'x'.

So now we can introduce the following definition:

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 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. So 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 above 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. Therefore T is inconsistent, contradicting our initial assumption that T is consistent.

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

So, if T is properly formalized, consistent and can prove enough arithmetic, then there is no way of mechanically determining what's a T-theorem and what isn't.

4.4 Diagonalization

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

Well, just imagine the square array you get by writing $\varphi_0(\overline{0})$, $\varphi_0(\overline{1})$, $\varphi_0(\overline{2})$, etc. in the first row, $\varphi_1(\overline{0})$, $\varphi_1(\overline{1})$, $\varphi_1(\overline{2})$, etc. in the next row, $\varphi_2(\overline{0})$, $\varphi_2(\overline{1})$, $\varphi_2(\overline{2})$ etc. in the next row, and so on. *Then 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 construction. And it is a diagonalization of this kind that is really at the heart of Gödel's incompleteness proof.[2]

4.5 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 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 unproved lemmas along the way).[3]

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. Theorem 2 claimed incompleteness on the assumption that T can prove a certain as-yet-unspecified amount of arithmetic. Our new Theorem 8 claims incompleteness on the more specific basis that, for any decidable property of numbers, T can case-by-case determine which numbers have the property. 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 project here: we will want to give a sharper account of what makes for an effectively decidable

[2]For the grandfather of all diagonalization arguments, due to Georg Cantor, see http://en.wikipedia.org/wiki/Cantor's_diagonal_argument (as well as *IGT2*, §2.5).

[3]I learnt the argument in this chapter as a student – so decades ago! – from lectures by Timothy Smiley.

property in order to (i) clarify the notion of sufficient strength, while (ii) still making it plausible that we want sufficiently strong theories in this clarified sense.

That can indeed be done, and it turns out that a surprisingly weak theory called Robinson Arithmetic which we meet in the next chapter is already sufficiently strong. However, supplying and defending the needed sharp account of the notion of effective decidability in order to pin down the notion of sufficient strength takes some effort! And it arguably takes at least as much effort compared with the task of filling in the needed details for proving incompleteness by Gödel's original method as partially sketched in Chapter 3. So over the next chapters, we are going to revert to exploring something closer to Gödel's route to the incompleteness theorems.

Still, our argument in this present chapter is highly suggestive and well worth knowing about.

5 Two weak arithmetics

Theorem 8 tells us that if a consistent, effectively axiomatized, theory is 'sufficiently strong', then it must be negation incomplete. And this isn't a wildly extravagant requirement, being sufficient strong: as we announced in §4.5, even the very weak arithmetic called Robinson Arithmetic already meets the condition (in which case, stronger arithmetics must also meet the condition). But we didn't say anything at all about what this weak theory looks like! In fact, we haven't looked at *any* detailed theory of arithmetic yet. It is obviously high time that we stop operating at the level of abstraction of earlier chapters; we need to start getting our hands dirty.

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 (as a warm-up) and then the important Robinson Arithmetic. You can by all means skip fairly lightly over a few of the boring proof details here; but you do need to get a sense of how these two simple formal theories work, in preparation for the next chapter where we introduce the canonical first-order theory of arithmetic, Peano Arithmetic.

5.1 The language L_B

First we describe *the language of baby arithmetic*, L_B. Its symbols are

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

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 §1.5). Recall, we use '$\overline{\mathsf{n}}$' to represent the standard numeral $\mathsf{SS}\ldots\mathsf{S0}$ with n occurrences of 'S'. Thus '$\overline{\mathsf{3}}$' is short for '$\mathsf{SSS0}$' which denotes the number 3.

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 '0' and applications of the function expressions 'S', '+', '×'. So, 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 the likes of τ as metalinguistic placeholders for terms of L_B. 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 logical identity sign. Since L_B lacks other 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 abbreviate that last wff as $\mathsf{S0} + \mathsf{SS0} \neq \mathsf{SS0} \times \mathsf{SS0}$.[1]

5.2 The axioms and logic of Baby Arithmetic

The theory BA couched in this language L_B will come equipped with a classical propositional deductive system to deal with negation and identity. You can choose your favourite system. In illustrations, we'll set out proofs in a Fitch-like natural deduction format (because it is likely to be familiar, and is easy to follow if it isn't familiar): but absolutely nothing hangs on the choice.

Next, we want non-logical axioms governing the successor function. We want to capture the ideas 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 $m = n$. And 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 to say: *any sentence that you get from one of the following schemas by substituting standard numerals for the placeholders 'ζ', 'ξ' is an axiom.*

[1] Fine print: we can allow an equation to be proceeded by any number of negation signs, with the result counting as a (non-atomic) wff. But we are going to give L_B a classical logic, so in fact a pair of adjacent negation signs just cancel each other out. Hence we need only worry about atomic formulae and their negations. It wouldn't make much odds if we also gave L_B the other connectives. But it is crucial that L_B lacks the apparatus of quantification.

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

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

NB: These schemas aren't 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.[2] We'll see some examples in a moment.

Next, we want non-logical axioms for addition. This time we want to capture the idea that adding zero to a number makes no difference: for any m, $m+0 = m$. And 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. $m + (n + 1) = (m + n) + 1$. Those 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.

Note that because of L_B's lack of quantifiers, we again can't express all that directly inside L_B itself. We again 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 numerical 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 with 'ξ' replaced by 'S0' to tell us how to multiply by two (multiply by one and then do some addition). And so on and so forth, thus defining multiplication for every number.

To summarize, then,

Defn. 20. BA *is the theory whose language is L_B, whose logic comprises classical negation rules plus standard identity rules, and whose non-logical axioms are every numerical instance of Schemas (1) to (6).*

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

Note: BA has an infinite number of axioms – since *any* instance of our schemas counts as an axiom. However, although it isn't *finitely* axiomatized, it is still an *effectively* axiomatized theory: given a candidate wff, we can effectively decide whether it is an instance of one of those six schemas and hence an axiom.

5.3 Some proofs inside BA

Let's start with three brisk examples of how arithmetic can be done inside BA.

First, let's first show that $\mathsf{BA} \vdash 0 + \overline{2} = \overline{2}$. In other words, $0 + \mathsf{SS0} = \mathsf{SS0}$ is a theorem. Note carefully, this wff *isn't* an instance of Schema 3, so we have to do a bit of work to derive it:

1.	$0 + 0 = 0$	Axiom, instance of Schema 3
2.	$0 + \mathsf{S0} = \mathsf{S}(0 + 0)$	Axiom, instance of Schema 4
3.	$0 + \mathsf{S0} = \mathsf{S0}$	From 1, 2 by LL
4.	$0 + \mathsf{SS0} = \mathsf{S}(0 + \mathsf{S0})$	Axiom, instance of Schema 4
5.	$0 + \mathsf{SS0} = \mathsf{SS0}$	From 3, 4 by LL

('LL' of course indicates the use of Leibniz's Law, which allows us to substitute terms which are equal.)

Exactly similarly, we can prove $\overline{2} + \overline{2} = \overline{4}$, i.e. $\mathsf{SS0} + \mathsf{SS0} = \mathsf{SSSS0}$:

1.	$\mathsf{SS0} + 0 = \mathsf{SS0}$	Axiom, instance of Schema 3
2.	$\mathsf{SS0} + \mathsf{S0} = \mathsf{S}(\mathsf{SS0} + 0)$	Axiom, instance of Schema 4
3.	$\mathsf{SS0} + \mathsf{S0} = \mathsf{SSS0}$	From 1, 2 by LL
4.	$\mathsf{SS0} + \mathsf{SS0} = \mathsf{S}(\mathsf{SS0} + \mathsf{S0})$	Axiom, instance of Schema 4
5.	$\mathsf{SS0} + \mathsf{SS0} = \mathsf{SSSS0}$	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 $\mathsf{SS0} \times \mathsf{SS0} = \mathsf{SSSS0}$:

1.	$\mathsf{SS0} \times 0 = 0$	Axiom, instance of Schema 5
2.	$\mathsf{SS0} \times \mathsf{S0} = (\mathsf{SS0} \times 0) + \mathsf{SS0}$	Axiom, instance of Schema 6
3.	$\mathsf{SS0} \times \mathsf{S0} = 0 + \mathsf{SS0}$	From 1, 2 by LL
4.	$0 + \mathsf{SS0} = \mathsf{SS0}$	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 $\overline{\mathsf{m}} + 0 = \overline{\mathsf{m}}$, 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. $\overline{\mathsf{m}} + \overline{2} = \overline{\mathsf{m} + 2}$.

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 $\overline{\mathsf{m}} + \overline{3} = \overline{\mathsf{m} + 3}$, and $\overline{\mathsf{m}} + \overline{4} = \overline{\mathsf{m} + 4}$, and so on. In fact, for any m, n, $\mathsf{BA} \vdash \overline{\mathsf{m}} + \overline{\mathsf{n}} = \overline{\mathsf{m} + \mathsf{n}}$.

Similarly, looking at the second proof pattern, we see that we'll be able to similarly prove $\overline{\mathsf{m}} \times \overline{2} = \overline{\mathsf{m} \times 2}$ 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 $\overline{\mathsf{m}} \times \overline{3} = \overline{\mathsf{m} \times 3}$, and $\overline{\mathsf{m}} \times \overline{4} = \overline{\mathsf{m} \times 4}$, and so on. In fact, take any m, n: then $\mathsf{BA} \vdash \overline{\mathsf{m}} \times \overline{\mathsf{n}} = \overline{\mathsf{m} \times \mathsf{n}}$.

We can now generalize a step further: BA can in fact 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 $(\overline{2} + \overline{3}) \times (\overline{2} \times \overline{2}) = \overline{20}$ – you'll forgive me for not writing out '$\overline{20}$' in basic notation with its twenty occurrences of 'S'! Then we can proceed as follows.

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

Next, we note that BA knows that different standard numerals are indeed not equal. For example, let's show that $\mathsf{BA} \vdash \overline{4} \neq \overline{2}$.

1.	$\mathsf{SSSS0} = \mathsf{SS0}$	Supposition
2.	$\mathsf{SSSS0} = \mathsf{SS0} \rightarrow \mathsf{SSS0} = \mathsf{S0}$	Axiom, instance of Schema 2
3.	$\mathsf{SSS0} = \mathsf{S0}$	From 1, 2 by MP
4.	$\mathsf{SSS0} = \mathsf{S0} \rightarrow \mathsf{SS0} = 0$	Axiom, instance of Schema 2
5.	$\mathsf{SS0} = 0$	From 3, 4 by MP
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 RAA.

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

Theorem 10. *If s and t are distinct numbers, then* $\mathsf{BA} \vdash \overline{\mathsf{s}} \neq \overline{\mathsf{t}}$.

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

We can immediately conclude from our last two theorems that

Theorem 11. *For any L_B wff φ, if φ is true then* BA $\vdash \varphi$*, and if φ is false then* BA $\vdash \neg\varphi$*. In a phrase,* BA correctly decides *every L_B wff.*

Proof. The only wffs of BA are equations preceded by zero or more negation signs. Since our logic is classical, we can ignore pairs of negation signs, so any wff φ is equivalent to either (a) $\sigma = \tau$ or (b) $\sigma \neq \tau$, for some terms σ, τ. Let σ evaluate to s and τ evaluate to t. Then by Theorem 9, (i) BA $\vdash \sigma = \overline{\mathsf{s}}$ and (ii) BA $\vdash \tau = \overline{\mathsf{t}}$.

And now we just consider the four possible cases.

(a) Suppose φ is equivalent to $\sigma = \tau$, and is true because $s = t$. Then $\overline{\mathsf{s}}$ must be the very same numeral as $\overline{\mathsf{t}}$. We can therefore immediately conclude from (i) and (ii) that BA $\vdash \sigma = \tau$ by the logic of identity. So BA $\vdash \varphi$.

(b) Suppose alternatively that φ is equivalent to $\sigma = \tau$, and is false because $s \neq t$. Then by Theorem 10, BA $\vdash \overline{\mathsf{s}} \neq \overline{\mathsf{t}}$, and together with (i) and (ii) that implies BA $\vdash \sigma \neq \tau$, again by the logic of identity. So BA $\vdash \neg\varphi$.

(c) Suppose φ is equivalent to $\sigma \neq \tau$, and is true because $s \neq t$. Then as in (b), BA $\vdash \sigma \neq \tau$. So now BA $\vdash \varphi$.

(d) Suppose finally that φ is equivalent to $\sigma \neq \tau$ and is false because $s = t$. As in (a), BA $\vdash \sigma = \tau$, hence BA $\vdash \neg\varphi$.

Hence, L_B wff correctly decides every φ. □

We therefore immediately have

Theorem 12. BA *is a sound effectively axiomatized theory which is negation complete.*

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 effectively axiomatized. And we've just seen that for every wff φ, it proves the true one of φ and $\neg\varphi$; and so is a negation complete theory.

"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 the language makes all the difference?" Exactly!

5.5 Robinson Arithmetic, Q

That's all very straightforward, but also pretty unexciting. The reason that Baby Arithmetic manages to prove every correct claim that it can express – and is

therefore negation complete by our Defn. 8 – 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 start beefing up BA into something more expressively competent is to restore the familiar apparatus of quantifiers and variables. That's what we'll start doing.

First, then, we define *the first-order language of basic arithmetic* L_A (compare §1.5). We'll 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 linguistic resources of first-order logic, all the connectives plus the usual supply of quantifiers and variables to 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 (so this is the least ambitious language which 'contains the language of basic arithmetic' in the sense of Defn. 11).

Now for *Robinson Arithmetic*, commonly denoted simply 'Q'. This is a theory built in the formal language L_A, and coming equipped with a full first order proof system for classical logic. And as 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. $\forall \mathsf{x}(0 \neq \mathsf{Sx})$

Axiom 2. $\forall \mathsf{x}\forall \mathsf{y}(\mathsf{Sx} = \mathsf{Sy} \rightarrow \mathsf{x} = \mathsf{y})$

Obviously, each instance of our earlier Schemas 1 and 2 can be deduced from the corresponding 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 '$0, \mathsf{S}0, \mathsf{SS}0, \ldots$'. But now we have the quantifiers in play. And these quantifiers are intended to run over the natural numbers; we certainly don't intend them to be also running over stray objects that aren't successors and aren't zero either. So let's reflect that in our axioms by explicitly ruling out strays. Other than zero, every number is a successor:

Axiom 3. $\forall \mathsf{x}(\mathsf{x} \neq 0 \ \rightarrow \ \exists \mathsf{y}(\mathsf{x} = \mathsf{Sy}))$

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

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

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

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 axioms entails all the instances of BA's corresponding schema. In sum, then:

Defn. 21. *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.

It is worth noting, for future reference, that Q was first isolated as a weak system of arithmetic worthy of study by Raphael M. Robinson in 1952 – i.e. long after Gödelian incompleteness was discovered.

5.6 Robinson Arithmetic is not complete

Like BA, Q too is assuredly an effectively axiomatized sound theory. Its axioms are all true; and its logic is truth-preserving; so its derivations are proper 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 13. Q *correctly decides every quantifier-free L_A sentence. In other words,* $Q \vdash \varphi$ *if the quantifier-free wff φ is true, and* $Q \vdash \neg\varphi$ *if the quantifier-free wff φ is false.*

Proof. We know that Q like BA will correctly decide every atomic wff, i.e. every equation between terms. We can then appeal to a background result of propositional logic which tells us that Q must then correctly decide every wff but up from those atoms using just the propositional connectives.[3] □

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*:

[3] If you insist on details: Suppose the wff φ contains the atoms $\alpha_1, \alpha_2, \ldots, \alpha_n$, and consider any assignment V of truth-values to those atoms. Let α_n^V be α_n if that is true on V, and $\neg\alpha_n$ otherwise. Similarly, let φ^V be φ if that is true on V, and $\neg\varphi$ otherwise.

Consider the line of a truth table for φ corresponding to V. On the left of the table, this line assigns values to the atoms, and tells us that $\alpha_1^V, \alpha_2^V, \ldots, \alpha_n^V$ are all true. The corresponding assignment of a value to φ tells us that φ^V true. And the background result from propositional logic that we need is that $\alpha_1^V, \alpha_2^V, \ldots, \alpha_n^V \vdash_{PL} \varphi^V$.

Roughly, then, propositional logic can prove what's said by the line of a truth table for φ corresponding to the valuation V.

Now, a quantifier-free L_A sentence φ is built up using propositional connectives from atoms α of the form $\sigma = \tau$. Consider the valuation V that assigns these atoms their actual values. As in the proof of Theorem 11, Q proves the true atoms and proves the negations of the false ones, so Q proves each α^V. Hence by the background result it also proves φ^V, i.e. proves whichever is the true one of φ and $\neg\varphi$.

Theorem 14. $\mathsf{Q} \nvdash \forall \mathsf{x}(\mathsf{0} + \mathsf{x} = \mathsf{x})$.

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). Hence one way of showing that $\mathsf{Q} \nvdash \varphi$ is to show that $\mathsf{Q} \nvDash \varphi$: 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 φ false.

So here goes: take the domain of our deviant, unintended, re-interpretation to be the set N^* comprising 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 '$\mathsf{0}$' still to 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 a successor (perhaps a self-successor!).

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 matrix (read off *row* $+^*$ *column*):

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

It is again easily checked that interpreting '+' as addition* still makes Axioms 4 and 5 true. (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) Finally, $x +^* S^*b = S^*(x +^* b)$. Which covers every possibility.)

We are not quite done, however, as we still need to show that we can give a co-ordinate re-interpretation of '$\times$' 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 an overall interpretation which makes the axioms of Q true and $\forall \mathsf{x}(\mathsf{0} + \mathsf{x} = \mathsf{x})$ false. So $\mathsf{Q} \nvdash \forall \mathsf{x}(\mathsf{0} + \mathsf{x} = \mathsf{x})$ □

Theorem 15. Q *is negation incomplete.*

Proof. Put $\varphi = \forall \mathsf{x}(\mathsf{0} + \mathsf{x} = \mathsf{x})$. We've just shown that $\mathsf{Q} \nvdash \varphi$. But obviously, Q can't prove $\neg\varphi$ either. Just revert to the standard interpretation built into L_A. Q

certainly has true axioms on this interpretation. So all theorems are true on that interpretation, but $\neg\varphi$ is false on that interpretation, so it can't be a theorem. Hence φ 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 as elaborate as Gödel's arguments to see that Q is incomplete. Q is, so to speak, *boringly* incomplete.

5.7 Statements of order in Robinson Arithmetic

Let's now start thinking about the properties and relations that be captured in Robinson Arithmetic – recalling the definition of §4.2.

Here's an example that will be useful:

Theorem 16. *In Robinson Arithmetic, the* less-than-or-equal-to *relation is not just expressed but captured by the wff* $\exists\mathsf{v}(\mathsf{v}+\mathsf{x}=\mathsf{y})$.

It is obvious that the wff expresses the relation. So what we need to 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 $\mathsf{Q} \vdash \neg\exists\mathsf{v}(\mathsf{v}+\overline{\mathsf{m}}=\overline{\mathsf{n}})$.

Proof 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.	$\quad\mathsf{a}+\mathsf{SS0}=\mathsf{S0}$	Supposition
3.	$\quad\mathsf{a}+\mathsf{SS0}=\mathsf{S}(\mathsf{a}+\mathsf{S0})$	From Axiom 5
4.	$\quad\mathsf{S}(\mathsf{a}+\mathsf{S0})=\mathsf{S0}$	From 2, 3 by LL
5.	$\quad\mathsf{a}+\mathsf{S0}=\mathsf{S}(\mathsf{a}+0)$	From Axiom 5
6.	$\quad\mathsf{SS}(\mathsf{a}+0)=\mathsf{S0}$	From 4, 5 by LL
7.	$\quad\mathsf{a}+0=\mathsf{a}$	From Axiom 4
8.	$\quad\mathsf{SSa}=\mathsf{S0}$	From 6, 7 by LL
9.	$\quad\mathsf{SSa}=\mathsf{S0}\rightarrow\mathsf{Sa}=0$	From Axiom 2
10.	$\quad\mathsf{Sa}=0$	From 8, 9 by MP
11.	$\quad0=\mathsf{Sa}$	From 10
12.	$\quad0\neq\mathsf{Sa}$	From Axiom 1
13.	$\quad$Contradiction!	From 11, 12
14.	Contradiction!	∃E 1, 2–13
15.	$\neg\exists\mathsf{v}(\mathsf{v}+\mathsf{SS0}=\mathsf{S0})$	RAA 1–14

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 done the 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 result we've just proved, we can sensibly add the standard symbol '$\leq$' to L_A, the language of Q, defined so that whatever terms – not just numerals – we put for the placeholders 'ξ' and 'ζ', $\xi \leq \zeta$ is just short for $\exists\mathsf{v}(\mathsf{v} + \xi = \zeta)$, and then Q will be able to prove at least the expected facts about the less-than-or-equals relations among quantifier-free terms. (Well, we really need to be a bit more careful than that in stating the rule for unpacking the abbreviation, if we are to avoid any possible 'clash of variables'. But we're not going to fuss about that sort of detail.)

Note, by the way, that some presentations in fact 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.[4] Since it so greatly helps readability, we'll henceforth make very free use of '$\leq$' as an abbreviatory symbol inside formal arithmetics.

5.8 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, despite its great shortcomings, Q does have some nice properties.

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 announce a quite general result:

Theorem 17. Q *can capture* all *decidable numerical properties – i.e. it is sufficiently strong in the sense of Defn 19.*

That might initially seem very surprising indeed, 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 proving enough facts about *particular* numbers.

So that's why Q is especially interesting – it is about the weakest arithmetic which is sufficiently strong (and it was isolated by Robinson for just that reason), and so about the weakest arithmetic for which Gödelian proofs of incompleteness can be run. Suppose, then, that a theory is formally axiomatized, consistent and

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

can prove everything Q can prove (those do indeed seem very modest 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 17 that Q *does* have sufficient strength if and when we have a quite general theory of effective decidability to hand. And we don't want to get embroiled in developing that theory (at least just yet). So what we *will* be proving quite soon (in Chapter 9) is a somewhat weaker claim about Q. We'll show that it can capture all so-called 'primitive recursive' properties, where these form a large and very important subclass of the decidable properties. 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.