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 very 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 of Advanced Studies at Princeton.

Gödel's doctoral dissertation, written when he was 23, established the *completeness* theorem for the first-order predicate calculus (i.e. 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 timeline curves' (where, in some sense, time travel is possible). Always a perfectionist, after the mid 1940s he more or less stopped publishing.

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 is *'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.*

To put it another way, 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.)

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

So we could say 'computably decidable' instead of 'effectively decidable'. Relatedly, we can talk e.g. about effectively determining what the main connective of a sentence is, meaning we have an algorithmic procedure a computer could use to find out what the main connective is; and so on.

How satisfactory is our definition, 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)?

Well, it turns out that the notion of effective decidability is 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 suitably sharpened to make the idea of effective decidability in good shape.

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, 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 logic books 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. (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*!)

[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.

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.

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 given purported 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.*

Defn. 5. *'$T \vDash \varphi$' 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 '$\vDash$' signifies *semantic entailment*, a semantic relation defined by generalizing over (re)interpretations of the relevant vocabulary.

Of course, we normally want a formal deduction to indeed be genuinely truth-preserving; so we will want our proof system to respect semantic 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 proved 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 *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 *complete theory*. For example T can't decide whether q is true. And there are lots of other wffs φ such that we have neither $T \vdash \varphi$ nor $T \vdash \neg\varphi$.

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 k, if there is one, such that $m + k = n$. 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*.[2] 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.

[2] *Principia* without the dodgy Axiom and without the Axiom of Infinity is one version of a so-called 'type theory' which is then quite nicely motivated, but sadly you can't reconstruct much maths in it.

[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.