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 in 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. Similarly 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 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 an L-expression or sequence of L-expressions 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 T, and suppose we are given a Gödel-numbering scheme. Then the corresponding properties/relation* 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 anon.

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 is, if necessary, a restricted quantifier running just over 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: G_T doesn't *really* say that – G_T is just a fancy sentence in the language of basic arithmetic, so is in fact just about *numbers*). Still, with that point in mind, you'll probably immediately spot that we can now prove ...

3.7 Incompleteness!

Theorem 1. *If T is a sound formal axiomatized theory whose language contains the language of basic arithmetic, then 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 equivalent to 'G_T' *is not provable in T*, and then such a sentence can neither be proved or 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 is equivalent to '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.