# Interlude

Here again are the two principal versions of the First Theorem which we proved. First, a theorem with a semantic assumption:

**Theorem 37.** *If $T$ is a sound p.r. axiomatized theory whose language contains the language of basic arithmetic, then there will be a true $\Pi_1$ sentence $\mathsf{G}_T$ such that $T \nvdash \mathsf{G}_T$ and $T \nvdash \neg\mathsf{G}_T$, so $T$ is negation incomplete.*

Second, a theorem requiring instead a syntactic assumption:

**Theorem 44.** *If $T$ is a consistent p.r. axiomatized theory which contains $\mathsf{Q}$, then there will be a $\Pi_1$ sentence $\mathsf{G}_T$ such that $T \nvdash \mathsf{G}_T$, and if $T$ is $\omega$-consistent, $T \nvdash \neg\mathsf{G}_T$. Hence, assuming $\omega$-consistency and so consistency, $T$ is negation incomplete.*

We later saw that, thanks to Rosser's theorem, we can in fact strengthen the second of these theorems by dropping the additional assumption of $\omega$-consistency. We also emphasized the sense in which both of these theorems can be better thought of as *incompletability* theorems.

Before we move on to introduce the Second Theorem in the next chapter, this Interlude pauses to quickly review three issues we have left dangling. We have already promised to say something about the first issue: how to beef up these stated Theorems (which talk of p.r. axiomatized theories) in order to establish versions of our originally announced Theorems 1 and 2 (which talked of effectively axiomatized theories more generally). Secondly, we need to revisit the lovely Theorem 7 which told us that consistent, effectively axiomatized, sufficiently strong theories are undecidable. A third issue may very well have occurred to you. We can raise it like this: "Ok, PA – to take that example of a nicely axiomatized rich theory of arithmetic – isn't complete. But the only formally undecidable sentences that we have met so far are strange Gödelian beasts, the results of decidedly unnatural coding constructions. Are there examples of *natural* arithmetic propositions, propositions we might already be mathematically interested in for their own sake, which are independent of PA?"

Let's take these issues in turn.

The primitive recursive functions are the numerical functions that can be computed without open-ended searches. So to get a more inclusive class of computable functions, the obvious move is to allow such searches. The standard way of implementing this move is by introducing a minimization operator, where $\mu x\varphi(x)$ returns the least number which satisfies the condition $\varphi$. Then we compute $\mu x\varphi(x)$ by checking in turn whether $\varphi(0)$, $\varphi(1)$, $\varphi(2)$, $\ldots$, searching until we get a positive result.[1]

Let's say that a function is *recursive* if it is a (total) function that can be defined from our stock of initial functions by applications of composition, primitive recursion, and/or minimization (compare Defn. 32). The recursive functions, then, are again effectively computable; and the primitive recursive functions are a subclass of them. As you would expect, we will also say a property is recursive if its characteristic function is recursive.

Note next that the $\mu$ operator is a kind of definite description operator. And we all know how to deal with definite descriptions in a standard first-order language using existential quantifiers and identity (use Russell's Theory of Descriptions!). Since we can therefore express the minimization operator using the logical apparatus of $L_A$, and we can already express composition and primitive recursion in $L_A$, it is therefore easy to strengthen the proof that (a language including) $L_A$ can express every primitive recursive function to show that it can express every recursive function. Similarly, we can strengthen the proof that (a theory including) Q can capture every primitive recursive function to show that it can capture every recursive function.

Now, a p.r. axiomatized theory $T$ is one for which e.g. the property of being an axiom is primitive recursive, and consequently the relation $Prf_T$ is primitive recursive. Likewise, a theory $T$ is recursively axiomatized if it is one for which e.g. the property of being an axiom is recursive and consequently the relation $Prf_T$ is recursive. Given that $L_A$ can express and Q can capture every recursive function, it is straightforward to check that our proofs of Theorems 37 and 44 will go through just as before if we now replace 'primitive recursive' with plain 'recursive', giving us strengthened Theorems.

All this is plain sailing. But where does it take us? True, our strengthened Theorems apply more widely.[2] However, we still haven't quite got back to Theorems 1 and 2, which talk of effectively axiomatized theories, not of recursively axiomatized theories. However, we can make the final connection if we accept what is known as *Church's Thesis*, which claims that the effectively computable numerical functions are in fact exactly the recursive functions.

Is this Thesis the sort of thing that can be *proved* correct, given the imprecision of the informal concept of effective computability? Perhaps not. But certainly, every serious attempt to pin down a sharply defined class of effective computable functions ends up locating the same class of recursive functions. For example, Alan Turing famously aimed to analyse the notion of computation and arrived

[1]What happens if no number satisfies $\varphi$? Good question. To keep things simple, assume for now that the $\mu$ operator is only applied to functions $\varphi$ such that $\exists x\varphi(x)$. That way, functions defined using the $\mu$ operator will remain total functions.

[2]But it is worth saying again that theories which are recursively axoimatized but not primitively recursively axiomatized will be odd – we don't normally regiment a theory in such a way that it will take an open-ended search to determine what's a axiom, for example!

at the definition of computability by a Turing machine: but we can quite easily prove that every numerical (total) function computable by a Turing machine is recursive, and vice versa. Because of results like this, Church's Thesis commands *very* wide support. Let's assume it. Then indeed we can recover Theorems 1 and 2.

Returning to Theorem 7, that invoked the notion of a 'sufficiently strong' theory which captures all effectively decidable numerical properties (see Defn. 19). By Church's Thesis, the effectively decidable numerical properties are the recursive ones. But $\mathsf{Q}$ can capture all recursive properties. So we can trade in the condition of being sufficiently strong for the condition of containing $\mathsf{Q}$. Similarly, we can trade in the notion of a theory's being effectively (un)decidable for the notion of its being recursively (un)decidable – meaning that the property of being the Gödel number of a theorem is (not) recursive. So then Theorem 7 becomes: a consistent, recursively axiomatized theory $T$ which contains $\mathsf{Q}$ is recursively undecidable. And this version has a quick proof. If the property of being a $T$ theorem were recursive, it would be capturable by $\mathsf{Prov}_T$, which it can't be by an obvious generalization of Theorem 51.

We know that there are computable functions which aren't primitive recursive from our diagonal argument for Theorem 24. But can we give a natural example of a computable, recursive-but-not-primitive-recursive function?

Here's one construction. Consider the two-place functions $f_1$, i.e. *sum* (repeated applications of the successor function), $f_2$, i.e. *product* (repeated sums), $f_3$, i.e. *exponentiation* (repeated products). As their respective arguments grow, the value of $f_1$ of course grows comparatively slowly, $f_2$ grows faster, $f_3$ faster still.

This sequence of functions can obviously be continued. Next comes $f_4$, the *super-exponential*, defined by repeated exponentiation:

$$x \Uparrow 0 = x$$
$$x \Uparrow Sy = x^{x \Uparrow y}.$$

Thus, for example, $3 \Uparrow 4$ is $3^{3^{3^{3^3}}}$ with a 'tower' of four exponents. Similarly, we can define $f_5$ (super-duper-exponentiation, i.e. repeated super-exponentiation), $f_6$ (repeated super-duper-exponentiation), and so on. The values of the respective functions $f_x$ grow faster and faster as their arguments are increased.

So now let's introduce the function $A(x) = f_x(x, x)$. The value of $A(x)$ grows *explosively*, running away ever faster as $x$ increases. Now, we can show that for any given one-place p.r. function $g(x)$ there is an $n$ such that its rate of growth as $x$ increases is no faster than $f_n(x, x)$. But $f_x(x, x)$ eventually grows faster than any particular $f_n(x, x)$, when $x > n$. Hence $A(x)$ isn't primitive recursive. Yet it is evidently computable, and with just a bit of work can be shown to be recursive.

Fast-growing functions are an old topic of mathematical interest, ever since G. H. Hardy wrote a classic paper about them in 1904. A fast-growing function like our function $A$ was first introduced by Wilhelm Ackermann in 1928 as an example of a computable, but not primitive recursive, function. $A$ is relatively tame, though, as fast-growing functions go. Like any recursive function, it can be expressed by an $L_A$ wff $\mathsf{A(x, y)}$; and – a further feature – $\mathsf{PA}$ 'knows' that $A$ is a total function, meaning that we have $\mathsf{PA} \vdash \forall \mathsf{x} \exists ! \mathsf{y} \mathsf{A(x, y)}$.

However, we can define wilder and wilder fast-growing functions, to the point where they lose that property of being provably-total-according-to-PA. For example, we can define the so-called Goodstein function $G$; this is a super-fast-growing recursive function which – like any recursive function – can be expressed in PA by an $L_A$ wff $\mathsf{G(x, y)}$. But this time, PA doesn't 'know' that $G$ is a total function, meaning that $\mathsf{PA} \nvdash \forall \mathsf{x} \exists ! \mathsf{y} \mathsf{G(x, y)}$.

It would take us too far afield to define the Goodstein function; and neither the proof that it really is a total function nor the proof that it isn't provably-total-according-to-PA is elementary. But without going into those details, we now have a gesture towards the answer to the question we raised at the beginning of this Interlude. Are there arithmetical truths of ordinary mathematical interest which are expressible in PA but undecidable by the theory? Yes, there is a whole family of such truths, truths about super-fast-growing functions.[3]

[3]For a lot more on the topics touched on in this Interlude, see *IGT2*. For example, Ch. 30 discusses Goodstein's Theorem and the Goodstein function. Ch. 38 discusses recursive functions in general and the Ackermann function in particular. Turing and Turing machines feature in Chs. 41 and 42. While Chs. 44 and 45 explore Church's Thesis.