Tennenbaum for Paula

Thomas Forster

March 1, 2017

O Tennenbaum, O Tennenbaum!
Wie treu sin' deine Blätter!!

...in which we exhibit a simplified weak version of Tennenbaum's theorem that requires less technical gadgetry. Specifically we shall prove:

THEOREM 1 *If $\mathfrak{M}$ is a nonstandard model of true arithmetic with carrier set $\mathbb{N}$ then the graphs of $+$ and $\times$ in $\mathfrak{M}$ cannot both be decidable.*

For these purposes *True Arithmetic* is the set of (first-order) sentences true in the standard model. The plan is to code up an undecidable set of standard naturals as a natural number of the model, and then use the decidability of the model to show that the undecidable set was decidable after all.

There are various ways of coding up (finite) sets of natural numbers as natural numbers. The cutest one known to me is the relation $n\,E\,m$ on natural numbers, defined to hold when the nth bit of m (m considered as a bit-string) is 1. (This is due to Wilhelm Ackermann). Clearly E is a decidable relation. It is useful to us because $\{m : m\,E\,n\}$ is an actual *set* (a subset of the model) and it is coded by the element n of the model.

Let T be a theory of arithmetic. Once we have erected a coding scheme we can use it to think of any nonstandard model $\mathfrak{M}$ of T as a structure for the language of second-order arithmetic, in the following way. $\mathfrak{M}$ has a standard part, and any nonstandard element of M has the potential to encode sets that are unbounded in the standard part. According to our E relation no two (nonstandard) naturals can encode the same set. However we are interested in the *standard part* of any set coded by a (nonstandard) natural, and—on this view—two distinct (nonstandard) naturals can encode the same set of *standard* naturals. The structure-for-the-language-of-second-order-arithmetic to which $\mathfrak{M}$ corresponds has as its carrier set the standard part of M. The range of the second order variables is the whole of M. Let us call this structure "$\mathfrak{M}^*$".

We start with a simple observation.

REMARK 1 *In $\mathfrak{M}^*$, every decidable set of standard naturals is encoded by a [nonstandard] natural [of $\mathfrak{M}$].*

Proof:

Let's write '$x \oplus y$' for the logical `or` of the two naturals x and y thought of as bit-strings. [There is nothing specific to $\mathfrak{M}^*$ here: this is happening in T]. Let $P()$ be a decidable predicate.

We will need the function f defined as follows.

$$\begin{aligned} f(0) &= 0; \\ f(n+1) &= \texttt{if}\ \ P(n+1)\ \ \texttt{then}\ \ f(n) \oplus 2^{n+1}\ \ \texttt{else}\ \ f(n). \end{aligned}$$

Now we find, in $\mathfrak{M}$, that $\{n : n \text{ is standard } \wedge P(n)\}$ is encoded by $f(m)$ whenever m is nonstandard (every nonstandard number is bigger than every standard number, after all) and $f(m)$ is of course a set of $\mathfrak{M}^*$. That is to say that, in $\mathfrak{M}^*$, any decidable set of standard natural numbers is a set of $\mathfrak{M}^*$—as desired. ■

Are there any *un*decidable predicates we can encode? If there are, then we can derive a contradiction from the assumption that the graph of + and × in the model is decidable (in a way that I shall spell out later). Now the obvious undecidable predicate is the halting set. Let us attempt to encode it and see what happens. Define f as follows:

$$\begin{aligned} f(0) &= 0; \\ f(n+1) &= \texttt{let}\ \ n+1 = \langle p, i, t\rangle\ \ \texttt{in} \\ &\qquad \texttt{if}\ \ \{p\}(i) \text{ halts within } t \text{ steps}\ \ \texttt{then}\ \ f(n) \oplus 2^{\langle p,i\rangle}\ \ \texttt{else}\ \ f(n). \end{aligned}$$

The key to understanding what f does is to fix some pair $\langle p, i\rangle$ in one's mind and think about what happens to the $\langle p, i\rangle$th bit of $f(n)$ as n gets bigger. It starts off clear, but gets set if $\{p\}(i)$ ever halts; and—once set—it remains set. The values of f are an ever-improving sequence of approximations to an E-style numerical code for the halting set, $\{\langle p, i\rangle : \{p\}(i)\Downarrow\}$.

So does $f(n)$ for some (indeed *any*) nonstandard n encode the halting set? You'd think so, but the catch is that $\mathfrak{M}$ might lie about whether or not $\{p\}(i)$ halts, by saying that it halts when in fact it doesn't. For example, $\mathfrak{M}$ could be a model of the theory $T = \text{PA} + \neg\text{Con(PA)}$. T is consistent but unsound. Since $\text{PA} \subseteq T$ and $T \vdash \neg\text{Con(PA)}$ then certainly $T \vdash \neg\text{Con}(T)$. T is recursively axiomatisable so the predicate "…is a T-proof" is decidable. Now let P be the program which, on being given an input i, examines [in order] all numbers bigger than i and stops when it finds one that encodes a T-proof of $\neg\text{Con}(T)$. P clearly halts on all standard inputs, and never halts in standard time, since all proofs of $\neg\text{Con}(T)$ are of nonstandard length.

> *It simply isn't true that a standard p, applied to a standard i, halts in standard time if it halts at all.*

So we need to put some restriction on $Th(\mathfrak{M})$ to forstall the possibility of $\mathfrak{M}$ thinking that there is a Turing machine whose gnumber is standard that takes a nonstandard number of steps to halt on some standard input. Now true arithmetic says this cannot happen. After all, if true arithmetic proves that there is an n s.t. $\{p\}(i)$ halts in n steps then there is some n such that true arithmetic proves that $\{p\}(i)$ halts in n steps, and this n is of course standard.

Thus, if $\mathfrak{M}$ is a nonstandard model of true arithmetic, it contains a number m s.t. for all standard naturals n of $\mathfrak{M}$, $\mathfrak{M} \models n\,E\,m$ iff n encodes a halting computation. If the graphs of = and $\times$ in $\mathfrak{M}$ are decidable, so is the graph of E. So suppose these graphs are decidable. We can now solve the Halting problem, as follows. You want to know whether or not $\{17\}$ halts on 23? Easy. You find 17 and 23 in $\mathfrak{M}$. They won't actually *be* 17 and 23 of course, but—since the property of being 17 (or 23) can be described using only '0' and the successor function—locating the actors who play those two parts in $\mathfrak{M}$ is straightforward: you examine the carrier set $\mathbb{N}$ of $\mathfrak{M}$ in increasing order until you find them. Then you locate the number p which $\mathfrak{M}$ believes to be the ordered pair of those two. That's do-able because pairing and unpairing are defined in terms of + and $\times$. Then you ask whether or not $\mathfrak{M}$ believes that $p\,E\,m$. That, too, is do-able, and for the same reasons.

■