-NoValue-

thy.1 The Halting Problem

cmp:thy:hlt: sec

By construction, the universal partial computable function $\mathrm{Un}(e, x)$ is defined if and only if the computation of the function coded by e produces a value for input x. It is natural to ask if we can decide whether this is the case. In fact, it is not. For the Turing machine model of computation, this means that whether a given Turing machine halts on a given input is computationally undecidable. The following theorem is therefore known as the "undecidability of the halting problem." We will provide two proofs below. The first continues the thread of our previous discussion, while the second is more direct.

cmp:thy:hlt: thm:halting-problem

Teorema thy.1. *Let*

$$h(e, x) = \begin{cases} 1 & \textit{if } \mathrm{Un}(e, x) \textit{ is defined} \\ 0 & \textit{otherwise.} \end{cases}$$

Then h is not computable.

Prova. Suppose h is computable. We show that this would let us define a universal computable function. Define

$$\mathrm{Un}'(e, x) = \begin{cases} \mathrm{Un}(e, x) & \text{if } h(e, x) = 1 \\ 0 & \text{otherwise.} \end{cases}$$

But now $\mathrm{Un}'(e, x)$ is a total function, and is computable if h is. For instance, we could define g using primitive recursion, by

$$g(0, e, x) \simeq 0$$
$$g(y + 1, e, x) \simeq \mathrm{Un}(e, x);$$

then

$$\mathrm{Un}'(e, x) \simeq g(h(e, x), e, x).$$

Since $\mathrm{Un}'(e, x)$ agrees with $\mathrm{Un}(e, x)$ wherever the latter is defined, Un' is universal for those partial computable functions that happen to be total. But this contradicts **??**. □

Prova. Suppose $h(e, x)$ were computable. Define the function g by

$$g(x) = \begin{cases} 0 & \text{if } h(x, x) = 0 \\ \uparrow & \text{otherwise.} \end{cases}$$

The function g is partial computable. For example, one can define it as $\mu y\ h(x, x) = 0$. So, for some e, $g(x) \simeq \mathrm{Un}(e, x)$ for every x. Is g defined at e? If it is, then, by the definition of g, $h(e, e) = 0$ (h can only take the value 0 if it is defined). By the definition of h, this means that $\mathrm{Un}(e, e)$ is undefined. By our assumption that $g(x) \simeq \mathrm{Un}(e, x)$ for every x, we have that $g(e)$ is undefined, a contradiction. On the other hand, if $g(e)$ is undefined, then $h(e, e) \neq 0$, and so $h(e, e) = 1$. It follows that $\mathrm{Un}(e, e)$ is defined. But since $g(x) \simeq \mathrm{Un}(e, x)$, then $g(e)$ would also be defined. Again, a contradiction. $\square$

explanation We can describe this argument in terms of Turing machines. Suppose there were a Turing machine H that takes as input a description of a Turing machine E and an input x, and decides whether or not E halts on input x. Then we could build another Turing machine G which takes a single input x, runs H to decide if the machine M_x with index x halts on input x, and does the opposite. In other words, if H reports that M_x halts on input x, G goes into an infinite loop, and if H reports that M_x doesn't halt on input x, then G just halts. Does G halt on its own index as input? The argument above shows that it does if and only if it doesn't—a contradiction. So our supposition that there is a such Turing machine H must be false.

Créditos das Imagens

Referências Bibliográficas