The Halting Function

The most famous example of a function that is not Turing-computable is the **Halting Function.** There are actually two different versions of the Halting Function.

The first, $H(n, m)$, is a function from pairs of natural numbers to natural numbers, and is defined as follows:

$$H(n, m) = \begin{cases} 1 \text{ if the } n\text{th Turning Machine halts when given input } m; \\ 0 \text{ otherwise.} \end{cases}$$

Consider, for example, the Turing Machine whose program is the following:

$$0 \quad \_ \quad \_ \quad r \quad 0$$

This is the 2310th Turing Machine according to our scheme, because:

$$2^{0+1} \cdot 3^{0+1} \cdot 5^{0+1} \cdot 7^{0+1} \cdot 11^{0+1} = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 = 2310$$

When given input 0 (i.e. the empty string), this machine will start moving to the right and never stop. So the the 2310th Turing Machine does not halt on input 0. So (H 2310, 0) = 0. In contrast, when given input $n(n > 0)$, our Turing Machine halts immediately, since it has no command line telling it what to do when reading a “1” in state 0. This means, in particular, that it halts given input 2310. So H(2310) = 1.

The second version of the Halting Function, $H(n)$, is a function from natural numbers to natural numbers. It is defined on the basis of the first version of the Halting Function:

$$H(n) = H(n, n)$$

So, for example, H(2310) = 1.

In what follows we’ll verify that $H(n)$ is not Turing-computable. (As you’ll be asked to verify in an exercise below, this entails that $H(n, m)$ is not Turing-computable either.) These results are due to Alan Turing.

Discussion

Hide Discussion

Topic: Week 9 / The Halting Function

Add a Post

Show all posts by recent activity

There are no posts in this topic yet.

✖