The Details

We are now in a position to prove our main result.

Gödel's Incompleteness Theorem
Let $\mathcal{L}$ be a language rich enough for the Lemma of Section 10.1.2 to be provable (for example, $\mathcal{L}$ might be our simple language L). Then no Turing Machine M is such that when run on an empty input:

1. M runs forever, outputting sentences of $\mathcal{L}$;
2. every true sentence of $\mathcal{L}$ is eventually output by M; and
3. no false sentence of $\mathcal{L}$ is ever output by M.

The proof proceeds by *reductio*.

We will assume that M is a Turing Machine that outputs all and only the true sentences of $\mathcal{L}$, and show that M's program can be used as a subroutine to construct a Turing Machine M^H which computes the Halting Function, which is impossible, since the Halting Function is not Turing-computable (as we saw in Chapter 9).

We will proceed in two steps. First, we will verify that if M existed, it could be used to construct M^H. We will then verify that M^H would compute the Halting Function, if it existed.

Step 1

Here is how to construct M^H, on the assumption that M exists. (Assume that M^H's input is a sequence of k ones.)

- M^H starts by running M's program as a subroutine. Each time M outputs a sentence, M^H proceeds as follows:

- If the sentence is "$\text{Halt}(k)$", M^H deletes everything on the tape, prints a one, and halts.
- If the sentence is "$\neg\text{Halt}(k)$", M^H deletes everything on the tape, and halts.
- Otherwise, M^H allows M to keep going.

Step 2

Let us verify that M^H would compute the Halting Function, if it existed. We will verify, in other words, that M^H outputs a one if the kth Turing Machine halts, and a zero otherwise.

- Suppose, first, that the kth Turing Machine halts given input k. Then our Lemma guarantees that "$\text{Halt}(k)$" is true. Since M will eventually output every true sentence of $\mathcal{L}$, this means that M will eventually output "$\text{Halt}(k)$". And since M will never output any falsehood, it will never output "$\neg\text{Halt}(k)$". So the construction of M^H guarantees that M^H will output a one.

- Now suppose that the kth Turing Machine does not halt given input k. Then our Lemma guarantees that "$\neg\text{Halt}(k)$" is true. Since M will eventually output every true sentence of $\mathcal{L}$, this means that M will eventually output "$\neg\text{Halt}(k)$". And since M will never output any falsehood, it will never print out "$\text{Halt}(k)$". So M^H will output a zero.

In summary, we have seen that if M existed, M^H would exist too. And we have seen that M^H would compute the Halting Function, which we know to be impossible. So M cannot exist.

This completes our proof of Gödel's Thoerem.

Discussion

Hide Discussion

Topic: Week 10 / A Proof of Godel's Theorem

Add a Post

Show all posts by recent activity

? Godel's theorem - no false sentence of L is ever output by M...
Hi, I am not able to see how this is equivalent to the other two versions of the theorem. Can som... 1