Some proofs written up

In case you had difficulties taking notes during the last lecture, I have written up all proofs I did on the OHP.

The diagonal function

Let $M_1, M_2, M_3, \ldots$ be an enumeration of Turing machines, and let $f_1, f_2, f_3, \ldots$ be the resulting enumeration of Turing-computable functions. The **diagonal function** d is defined as follows:

$$d(n) = \begin{cases} \perp & \text{if } f_n(n) \text{ is defined,} \\ 1 & \text{otherwise} \end{cases}$$

(Recall that we write $\perp$ for “undefined”.)

Uncomputability of d

Proposition. The diagonal function is not Turing-computable.

Proof. By contradiction. So suppose that d **is** Turing-computable. Then d is the n-th Turing-computable function for some n, i.e. $d = f_n$. We have

$$\begin{aligned} d(n) = 1 \iff & f_n(n) \text{ is undefined} && \text{(by definition of } d\text{)} \\ \iff & d(n) \text{ is undefined} && \text{(because } d = f_n\text{)} \end{aligned}$$

This is a contradiction, so d cannot be Turing-computable.

The halting function

The **halting function** is defined as follows:

$$h(n,k) = \begin{cases} 2 & \text{if } M_n \text{ halts on input } k \\ 1 & \text{otherwise} \end{cases}$$

Self-halting

The **self-halting function** is defined by $s(n) = h(n, n)$.

Proposition. The self-halting function s is not Turing-computable.

Proof (part 1 of 2)

By contradiction. Suppose that s is computable by some TM M. From M, we build new TM M' with the following property:

$$\begin{aligned} M' \text{ halts on input } n \quad & \text{iff } s(n) = 1 \\ M' \text{ does not halt on input } n \quad & \text{iff } s(n) = 2 \end{aligned}$$

Suppose we have M'. Then we get a contradiction as follows: we know that M' is the k-th TM for some k, i.e. $M' = M_k$. Now

$$\begin{aligned} M' \text{ halts on input } k \quad & \text{iff } M_k \text{ halts on input } k && \text{(because } M' = M_k\text{)} \\ & \text{iff } h(k,k) = 2 && \text{(by definition of } h\text{)} \\ & \text{iff } s(k) = 2 && \text{(by definition of } s\text{)} \\ & \text{iff } M' \text{ does not halt on input } k && \text{(because } M' \text{ has the above property)} \end{aligned}$$

This is a contradiction. So s cannot be Turing-computable.

On the next slide, we convince ourselves that M' can be built from the (hypothetical) TM

M.

Proof (part 2 of 2)

The machine M', on input n, first proceeds like M. Because M computes s, we know that M halts with configuration 1_q or 1_q1 for some state q (depending on whether $s(n)$ is 1 or 2.) Now M' checks whether there are one or two strokes on the tape. First, M' moves right, into some configuration 10_r or 11_r. In the case 10_r, M' halts. In the case 11_r, M' goes into an infinite loop $11_r \rightarrow 11_r \rightarrow 11_r \rightarrow \ldots$. The details of building M' (which I showed in the lecture last time) are straightforward.

Uncomputability of the halting function

Corollary. The halting function h is not Turing-computable.

Proof. The intuition behind this proof is simple: if the halting function h was computable, then the self-halting function s, being a "special case" of s, would also be computable. But s is not computable, so h is not computable either. Strictly speaking, we have to show that, if there was a TM M for h, then there would be a TM M' for s. But this is easy to see: M' works like M, except that it duplicates the initial block of 1's. E.g. if the initial tape is 11111, then M' produces the tape 11111011111 and proceeds like M. Building a TM for duplicating the initial block of 1's is easy and left as a (voluntary) exercise.