# CSE596, Fall 2008 Notes on Diagonalization Week 2⟶Week 4

Given an assignment of "Gödel numbers" to Turing machines to create a comprehensive list of programs labeled $M_0, M_1, M_2, \ldots$ Define the "diagonal language"

$$D_{\text{TM}} = \{ e : e \notin L(M_e) \}.$$

If $D_{\text{TM}}$ were computably enumerable (c.e.) then there would be a code $d$ of a TM $M_d$ such that $L(M_d) = D_{\text{TM}}$. But then

$$\begin{aligned} d \in D_{\text{TM}} &\iff d \in L(M_d) && \text{by } L(M_d) = D_{\text{TM}} \\ &\iff d \notin L(M_d) && \text{by definition of } D_{\text{TM}}, \end{aligned}$$

an unallowable contradiction. Thus $D_{\text{TM}}$ is not c.e.

This statement and proof can be regarded as a special case of the famous Diagonalization Theorem of Georg Cantor from the late 1800s, which in its purest form goes as follows:

**Cantor's Theorem.** For every set $S$, there is never a 1-1 correspondence $g$ between $S$ and its power set $\mathcal{P}(S) = \{ T : T \subseteq S \}$.

If $S$ is finite, $n = |S|$, then this is obvious, since $|\mathcal{P}(S)| = 2^n$ and $2^n$ is *always* greater than $n$. The point of the theorem is that the lack of a 1-1 correspondence applies to infinite sets too!

**Proof.** Given $g : S \longrightarrow \mathcal{P}(S)$, define $D = \{ e : e \text{ is not in the set } g(e) \}$. Clearly $D$ is a subset of $S$, i.e., $D \in \mathcal{P}(S)$. If $g$ were a 1-1 correspondence—or even just onto $\mathcal{P}(S)$—then there would be an element $d \in S$ such that $g(d) = D$. But then:

$$\begin{aligned} d \in D &\iff d \in g(d) && \text{by } g(d) = D \\ &\iff d \notin g(d) && \text{by definition of } D, \end{aligned}$$

the same contradiction! Take $S = \Sigma^*$ (or $S = \mathbf{N}$ if you view inputs as numbers rather than strings) and $g$ to be the mapping that associates to each Gödel number $e$ the language $L(M_e)$, and we get exactly the first proof.

Whereas lectures will stress that the "TM $M_d$" in the first proof is *unreal*, the function $g$ and the set $D$ in the second proof are very much real. Only the assumption of $d$ such that $g(d) = D$ is unreal. What we've actually proved is that every function $g : S \longrightarrow B$ (with $B = \mathcal{P}(S)$) fails to be onto $B$, since $D$ is never in its range. This is the *formal mathematical definition* of the cardinality of $B$ being greater than that of $S$, even for infinite sets (cf. Section 1.6 of the text). Cantor used the Hebrew first letter $\aleph$ for infinite cardinals, beginning with $\aleph_0 = |\mathbf{N}|$. Whether $|\mathcal{P}(\mathbf{N})|$ is the next-higher cardinal above $\aleph_0$ is an unsolved problem called the *Continuum Hypothesis*, which is known to be unresolvable within set theory itself. (We will re-visit diagonalization later in cases where "$M_0, M_1, M_2, \ldots$" is a major sub-sequence of the enumeration of Turing machines, and clever choice of this sub-sequence enables us to control the complexity of the resulting diagonal language.)

To see why Cantor's proof is called "diagonalization," let us visualize an infinite table whose rows are indexed by (Gödel numbers of) Turing machines, and whose columns are indexed by numbers (or strings).

| $x \in L(M_e)$? | 0 | 1 | 2 | 3 | 4 | 5 | 6 | … |
|---|---|---|---|---|---|---|---|---|
| $M_0$ | $\underline{0}$ | 0 | 0 | 0 | 0 | 0 | 0 | … |
| $M_1$ | 1 | $\underline{1}$ | 1 | 1 | 1 | 1 | 1 | … |
| $M_2$ | 1 | 0 | $\underline{1}$ | 0 | 1 | 0 | 1 | … |
| $M_3$ | 0 | 1 | 0 | $\underline{1}$ | 0 | 1 | 0 | … |
| $M_4$ | 0 | 0 | 1 | 1 | $\underline{0}$ | 1 | 0 | … |
| $M_5$ | 1 | 1 | 0 | 0 | 1 | $\underline{0}$ | 0 | … |
| ⋮ | … | | | | | | | |

The picture supposes that $L(M_0) = \emptyset$, $L(M_1) = \Sigma^*$, $M_2$ recognizes the even numbers, $M_3$ the odd numbers, $M_4$ the set of primes, $M_5$ the perfect squares, etc. Every row $e$ gives the sequence $s$ of values of the characteristic function of the language $L(M_e)$—i.e., the rows are languages that we can also call "$L_s$". If you flip-flop the values on the main diagonal, then you get exactly the characteristic sequence $1\ 0\ 0\ 0\ 1\ 1 \ldots$ of the diagonal language $D = \{\, e : e \notin L(M_e) \,\}$. Since this sequence differs from each row $d$ in the $d$th column, it cannot be a row in the table. It is true that the *particular* $D$ you get depends on the enumeration of TMs (i.e, the scheme for Gödel numbering, or in my parlance, the particular TM compiler) that you choose, but the upshot is always the same—*all* such languages $D$ are undecidable, indeed not r.e.

Turing's own take on the diagonalization diagram was to put a binary decimal point in front of each row sequence $s$, thus defining a real number $r = .s$ between 0 and 1. The correspondence from rows to reals fails to be 1-1 only for cases like $1/2 = .1000\ldots = .01111\ldots$ where the row language of the former sequence is finite, and these cases do not really bother the analysis. Turing observed that for *any* infinite binary sequence $s$ and $n \geq 1$, deciding whether the first $n$ strings/numbers belong to the corresponding language $L_s$ is the same task as computing $r = .s$ to $n$ binary decimal places. Thus a decidable language corresponds to a *computable number*. Since $\pi$ and $\sqrt{2}$ are computable, the languages given by their binary expansions are decidable. Thus the title of his famous paper, "On computable numbers, with application to the *Entscheidungsproblem*" (Ger. Decision Problem), ultimately refers to his proof that mathematics can define a number, namely $.D$, that it cannot compute.

Our text takes takes the "growth" rather than "flip-flop" view of diagonalization. If we allow arbitrary numbers in place of $0, 1$ as entries in the table, then each row $e$ gives the sequence of values $f_e(0), f_e(1), f_e(2), \ldots$ of a corresponding function $f_e : \mathbf{N} \longrightarrow \mathbf{N}$. If we take the main diagonal sequence and increase each of its values by 1, we obtain the sequence of values of the "diagonal function" $\delta(e) = f_e(e) + 1$. Since this sequence likewise differs from each row $d$ in the $d$th column, the function $\delta$ is not in the table. This shows that the set of functions from $\mathbf{N}$ to $\mathbf{N}$ has cardinality higher than $\aleph_0$, i.e. is *uncountable*. In fact, this set can be put in 1-1 correspondence with the set of languages.

If we re-define $\delta'(n) = 1 + \max\{\, f_e(x) : e, x \leq n \,\}$, then we get a diagonal function that actually grows faster than every function in the table. If we go all the way back to the row indices being Turing machines (or Java programs or RAM programs etc.), we run into the problem of entries $f_e(x) =$ the output of $M_e(x)$ being undefined when $M_e(x)$ doesn't halt. If $\mathrm{PT}_{\mathrm{TM}}$ (or even just the Halting Problem) were decidable, however, we could use the resulting procedure "TEST" to compute a substitute value 0 for those cases. The text's proof of Theorem 3.2 then yields the contradiction that $\delta'(n)$ then becomes computable, but doesn't appear in the table of computable functions. (The text actually uses $\delta'(n) = 1 + f_n(n)$.)