# Computational Complexity

## Lecture 4: Diagonalization and the Time Hierarchy Theorems

Ronald de Haan
me@ronalddehaan.eu

University of Amsterdam

February 12, 2026

# Recap

*What we saw last time..*

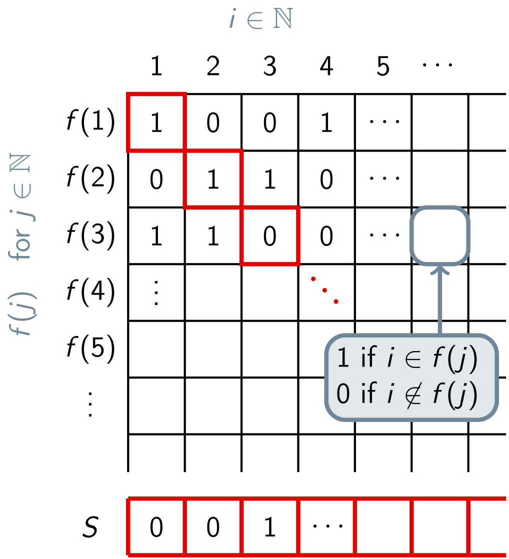
- Proof that NP-complete problems exist
- The Cook-Levin Theorem
- Concrete reductions between problems
- Search vs. decision problems

# What will we do today?

- Diagonalization arguments
- Time Hierarchy Theorems
- $\mathsf{P} \neq \mathsf{EXP}$

# Warm-up: Cantor's diagonal argument

- We show: $\mathcal{P}(\mathbb{N})$ is uncountable
- Suppose that it is countably infinite. Then there is some bijection $f : \mathbb{N} \to \mathcal{P}(\mathbb{N})$.
- Consider the set $S \in \mathcal{P}(\mathbb{N})$ such that for all $i \in \mathbb{N}$ it holds that $i \in S$ iff $i \notin f(i)$
- Then $S \neq f(i)$ for each $i \in \mathbb{N}$, so $f$ is not a bijection. $\lightning$



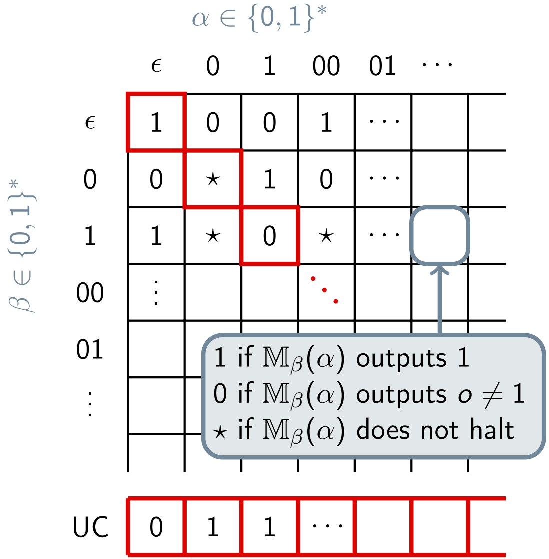
# Diagonalization over TMs: uncomputable functions

- We show that there exists an uncomputable function $\mathsf{UC} : \{0,1\}^* \to \{0,1\}$

- Define UC: for all $\alpha \in \{0,1\}^*$, $\mathsf{UC}(\alpha) = 0$, if $\mathbb{M}_\alpha(\alpha) = 1$, and $\mathsf{UC}(\alpha) = 1$ otherwise.

- Suppose that UC is computable. Then there exists some $\mathbb{M}_\beta$ that computes UC: $\mathbb{M}_\beta(\alpha) = \mathsf{UC}(\alpha)$ for all $\alpha \in \{0,1\}^*$.

- In particular, $\mathbb{M}_\beta(\beta) = \mathsf{UC}(\beta)$. By def. of UC: $\mathbb{M}_\beta(\beta) \neq \mathsf{UC}(\beta)$. $\lightning$



# Deterministic Time Hierarchy Theorem

**Theorem**

*If $f, g : \mathbb{N} \to \mathbb{N}$ are time-constructible functions such that $f(n) \log f(n)$ is $o(g(n))$, then $\mathsf{DTIME}(f(n)) \subsetneq \mathsf{DTIME}(g(n))$.*

- Assumption of time-constructibility rules out 'weird' functions.
  - $f$ is *time-constructible* if $f(n) \geq n$ and there exists a TM that computes the function $x \mapsto f(|x|)$ in time $O(f(|x|))$, for each $x \in \{0,1\}^*$
- We will prove $\mathsf{DTIME}(n) \subsetneq \mathsf{DTIME}(n^{1.5})$

# $\mathrm{DTIME}(n) \subsetneq \mathrm{DTIME}(n^{1.5})$

- Consider a TM $\mathbb{D}$ that, on input $\alpha \in \{0,1\}^*$, runs the simulation of $\mathbb{M}_\alpha(\alpha)$, and stops after $|\alpha|^{1.4}$ steps (counting the number of simulator steps), and:
  - if the simulation of $\mathbb{M}_\alpha(\alpha)$ outputs some $b \in \{0,1\}$ within $|\alpha|^{1.4}$ steps, then $\mathbb{D}(\alpha)$ outputs $1 - b$
  - otherwise, $\mathbb{D}(\alpha)$ outputs 1

diagonalization

- The language $L$ decided by $\mathbb{D}$ is in $\mathrm{DTIME}(n^{1.5})$
  - We perform a 'clocked' computation, maintaining a counter that keeps track of how many computation steps we took
  - Performing $T$ time steps of a computation (using such a counter) takes time $O(T \log T)$, and since $n^{1.4} \log n^{1.4}$ is $O(n^{1.5})$, we get that $L$ is in $\mathrm{DTIME}(n^{1.5})$
  - (This is where we need time-constructibility, for the general case: so that we can compute the number $T$ within $T$ time steps.)

# $\mathrm{DTIME}(n) \subsetneq \mathrm{DTIME}(n^{1.5})$

- Consider a TM $\mathbb{D}$ that, on input $\alpha \in \{0,1\}^*$, runs the simulation of $\mathbb{M}_\alpha(\alpha)$, and stops after $|\alpha|^{1.4}$ steps (counting the number of simulat**or** steps), and:
  - if the simulation of $\mathbb{M}_\alpha(\alpha)$ outputs some $b \in \{0,1\}$ within $|\alpha|^{1.4}$ steps, then $\mathbb{D}(\alpha)$ outputs $1 - b$
  - otherwise, $\mathbb{D}(\alpha)$ outputs 1

diagonalization

- We show that $L \notin \mathrm{DTIME}(n)$.
  - Suppose that $L \in \mathrm{DTIME}(n)$. Then there is some TM $\mathbb{M}$ that decides $L$ and runs in time $dn$, for some $d \in \mathbb{N}$.
  - Simulating $\mathbb{M}$ on input $x$ takes time $d'd|x|\log(d|x|)$, for some $d' \in \mathbb{N}$.
  - There is some $n_0 \in \mathbb{N}$ such that for all $n \geq n_0$ it holds that $n^{1.4} \geq d'dn\log(dn)$.
  - Let $\alpha$ be a string of length $\geq n_0$ that represents $\mathbb{M}$: $\mathbb{M} = \mathbb{M}_\alpha$
  - Then $\mathbb{M}_\alpha(\alpha) = \mathbb{D}(\alpha)$, because $\mathbb{M} = \mathbb{M}_\alpha$, and $\mathbb{M}$ and $\mathbb{D}$ decide the same language
  - The 'clocked' simulation of $\mathbb{M}_\alpha(\alpha)$ for $n^{1.4}$ steps finishes, because $n^{1.4} \geq d'dn\log(dn)$, and so $\mathbb{D}(\alpha) = 1 - \mathbb{M}_\alpha(\alpha) = 1 - \mathbb{D}(\alpha)$. $\lightning$

## P $\subsetneq$ EXP

- The functions $2^n$ and $2^{2n}$ are time-constructible, and $2^n \log 2^n = n \cdot 2^n$ is $o(2^{2n})$.
- Then by the Deterministic Time Hierarchy Theorem, $\mathrm{DTIME}(2^n) \subsetneq \mathrm{DTIME}(2^{2n})$.
- $\mathrm{P} = \cup_{c \in \mathbb{N}} \mathrm{DTIME}(n^c) \subseteq \mathrm{DTIME}(2^n) \subsetneq \mathrm{DTIME}(2^{2n}) \subseteq \mathrm{EXP}$
- So, $\mathrm{P} \neq \mathrm{EXP}$.

# Nondeterministic Time Hierarchy Theorem

**Theorem**

*If $f, g : \mathbb{N} \to \mathbb{N}$ are time-constructible functions such that $f(n+1)$ is $o(g(n))$, then* $\mathrm{NTIME}(f(n)) \subsetneq \mathrm{NTIME}(g(n))$.

- As a result: $\mathrm{NP} \subsetneq \mathrm{NEXP}$, where $\mathrm{NEXP} = \cup_{c \in \mathbb{N}} \mathrm{NTIME}(2^{n^c})$.

# Ladner's Theorem

- Question: is it the case that all problems in NP are either (i) in P or (ii) NP-complete?
- If $\mathsf{P} = \mathsf{NP}$, then this is trivially true.
- If $\mathsf{P} \neq \mathsf{NP}$, then no:

**Theorem (Ladner 1975)**

*Suppose that* $\mathsf{P} \neq \mathsf{NP}$*.*
*Then there exists a language* $L \in \mathsf{NP} \setminus \mathsf{P}$ *that is not* NP*-complete.*

- Proof uses a diagonalization argument.

# Recap

- Diagonalization arguments
- Time Hierarchy Theorems
- $\mathsf{P} \neq \mathsf{EXP}$

## Next time

- Can we use diagonalization to attack $\mathsf{P} \stackrel{?}{=} \mathsf{NP}$? (Spoiler: no.)
- Limits of diagonalization
- Relativizing results
- Oracles