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
- $P \neq EXP$
We show: \(\mathcal{P}(\mathbb{N})\) is uncountable.

Suppose that it is countably infinite. Then there is some bijection \(f : \mathbb{N} \rightarrow \mathcal{P}(\mathbb{N})\).

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.
We show that there exists an uncomputable function $UC : \{0, 1\}^* \rightarrow \{0, 1\}$.

Define $UC$: for all $\alpha \in \{0, 1\}^*$, $UC(\alpha) = 0$, if $M_\alpha(\alpha) = 1$, and $UC(\alpha) = 1$ otherwise.

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

In particular, $M_\beta(\beta) = UC(\beta)$. By def. of $UC$: $M_\beta(\beta) \neq UC(\beta)$. 

\[\begin{array}{ccccccc}
\varepsilon & 0 & 1 & 00 & 01 & \cdots \\
\hline
\varepsilon & 1 & 0 & 0 & 1 & \cdots \\
0 & 0 & * & 1 & 0 & \cdots \\
1 & 1 & * & 0 & * & \cdots \\
00 & \vdots & & & & \\
01 & \vdots & & & & \\
\vdots & & & & & \\
UC & 0 & 1 & 1 & \cdots \\
\end{array}\]
Deterministic Time Hierarchy Theorem

Theorem

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

- 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 $\text{DTIME}(n) \subsetneq \text{DTIME}(n^{1.5})$
Consider a TM $D$ that, on input $\alpha \in \{0, 1\}^*$, runs the simulation of $M_\alpha(\alpha)$, and stops after $|\alpha|^{1.4}$ steps (counting the number of simulator steps), and:

- if the simulation of $M_\alpha(\alpha)$ outputs some $b \in \{0, 1\}$ within $|\alpha|^{1.4}$ steps, then $D(\alpha)$ outputs $1 - b$
- otherwise, $D(\alpha)$ outputs 1

The language $L$ decided by $D$ is in $\text{DTIME}(n^{1.5})$

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 $\text{DTIME}(n^{1.5})$

(This is where we need time-constructibility, for the general case: so that we can compute the number $T$ within $T$ time steps.)
Consider a TM $D$ that, on input $\alpha \in \{0, 1\}^*$, runs the simulation of $M_\alpha(\alpha)$, and stops after $|\alpha|^{1.4}$ steps (counting the number of simulator steps), and:

- if the simulation of $M_\alpha(\alpha)$ outputs some $b \in \{0, 1\}$ within $|\alpha|^{1.4}$ steps, then $D(\alpha)$ outputs $1 - b$
- otherwise, $D(\alpha)$ outputs 1

We show that $L \notin \text{DTIME}(n)$.

- Suppose that $L \in \text{DTIME}(n)$. Then there is some TM $M$ that decides $L$ and runs in time $dn$, for some $d \in \mathbb{N}$.
- Simulating $M$ 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 $M$: $M = M_\alpha$
- Then $M_\alpha(\alpha) = D(\alpha)$, because $M = M_\alpha$, and $M$ and $D$ decide the same language
- The ‘clocked’ simulation of $M_\alpha(\alpha)$ for $n^{1.4}$ steps finishes, because $n^{1.4} \geq d'dn \log(dn)$, and so $D(\alpha) = 1 - M_\alpha(\alpha) = 1 - D(\alpha)$. \Footnote
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, $\text{DTIME}(2^n) \subsetneq \text{DTIME}(2^{2n})$.

$P = \bigcup_{c \in \mathbb{N}} \text{DTIME}(n^c) \subseteq \text{DTIME}(2^n) \subsetneq \text{DTIME}(2^{2n}) \subseteq \text{EXP}$

So, $P \neq \text{EXP}$. 
Theorem

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

As a result: \( \text{NP} \subsetneq \text{NEXP} \), where \( \text{NEXP} = \bigcup_{c \in \mathbb{N}} \text{NTIME}(2^{n^c}) \).
Ladner’s Theorem

Question: is it the case that all problems in NP are either (i) in P or (ii) NP-complete?

If $P = NP$, then this is trivially true.

If $P \neq NP$, then no:

Theorem (Ladner 1975)

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

Proof uses a diagonalization argument.
Recap

- Diagonalization arguments
- Time Hierarchy Theorems
- \( P \neq \text{EXP} \)
Next time

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