Recap

What we saw last time..

- Diagonalization arguments
- Time Hierarchy Theorems
- $P \neq EXP$
What will we do today?

- Can we use diagonalization to attack $P \neq \text{NP}$? (Spoiler: no.)
- Limits of diagonalization
- Relativizing results
- Oracles
One concrete interpretation of *diagonalization proofs*:

any proof technique that depends on the following properties of TMs:

1. effective representation of TMs by strings
2. ability of one TM to simulate another efficiently

We will see some limits of these proof techniques.
Oracles

- Black-box machine that can solve a decision problem $O$ in a single time-step
Definition

An oracle Turing machine is a TM $M$ that has a special (read-write) tape that we call the oracle tape and three special states $q_{\text{query}}, q_{\text{yes}}, q_{\text{no}} \in Q$.

To execute $M$, we specify some $O \subseteq \{0, 1\}^*$ that is used as the oracle for $M$.

Whenever during the execution, $M$ is in the state $q_{\text{query}}$ the machine (in the next step) enters the state $q_{\text{yes}}$ if $w \in O$ and the state $q_{\text{no}}$ if $w \notin O$—where $w$ denotes the current contents of the special oracle tape.

The tape contents and tape heads do not change/move.

$M^O(x)$ denotes the output of $M$ on input $x$ with oracle $O$.

An oracle TM knows how to use any oracle $O \subseteq \{0, 1\}^*$
Relativized complexity classes

Definition
Let \( O \subseteq \{0, 1\}^* \) be a decision problem.

- \( P^O \) is the set of all decision problems that can be decided by a polynomial-time deterministic TM with oracle access to \( O \).
- \( NP^O \) is the set of all decision problems that can be decided by a polynomial-time nondeterministic TM with oracle access to \( O \).

We will use similar notation for variants of other complexity classes that are based on Turing machines with bounds on the running time, e.g., \( EXP^O \).
One concrete interpretation of *diagonalization proofs*:

any proof technique that depends on the following properties of TMs:

(I) effective representation of TMs by strings

(II) ability of one TM to simulate another efficiently

We will see some limits of these proof techniques.
Relativizing results

- Regardless of the choice of $O \subseteq \{0, 1\}^*$, properties (I) and (II) also hold for oracle TMs.

- *Relativizing results* are results that depend only on (I) and (II)
  - E.g., $P \not\subseteq \text{EXP}$

- Relativizing results also hold when you add *any* oracle $O \subseteq \{0, 1\}^*$
  - E.g., $P^O \not\subseteq \text{EXP}^O$, for each $O \subseteq \{0, 1\}^*$
The Baker-Gill-Solovay Theorem

Theorem (Baker, Gill, Solovay 1975)

There exist $A, B \subseteq \{0, 1\}^*$ such that $P^A = NP^A$ and $P^B \neq NP^B$.

So no proof that $P = NP$ or $P \neq NP$ can be relativizing.
Let $A = \{ (\alpha, x, 1^n) \mid M_\alpha \text{ outputs 1 on input } x \text{ within } 2^n \text{ steps } \}$.

Then $\text{EXP} \subseteq P^A \subseteq \text{NP}^A \subseteq \text{EXP}$.

$\text{EXP} \subseteq P^A$ (idea):
- With one oracle query to $A$ you can do exponential-time computation in one step.

$\text{NP}^A \subseteq \text{EXP}$ (idea):
- Simulate computation of $\text{NP}^A$ machine in exponential time.
  - Enumerate all sequences of nondeterministic choices.
  - Compute answer to each (polynomial-size) oracle query.
Oracle $B$ such that $P^B \neq NP^B$

For any $B \subseteq \{0, 1\}^*$, let $U_B = \{ 1^n \mid \text{there is some } x \in \{0, 1\}^n \text{ such that } x \in B \}$.

Then $U_B \in NP^B$.

- On any input $1^n$, we use nondeterminism to guess $x \in \{0, 1\}^n$, and query the oracle $B$ to check if $x \in B$.

We construct some $B \subseteq \{0, 1\}^*$ such that $U_B \not\in P^B$.

- Using diagonalization. :-)

Construct $B \subseteq \{0, 1\}^*$ such that $U_B \notin P^B$

- We gradually build up $B$ in stages. Start with $\emptyset$. One stage for each $i \in \{0, 1\}^*$.

- In stage $i$:
  - For only finitely many strings $x$ we chose whether $x \in B$ or $x \notin B$. Let $n$ be larger than the length of any such $x$.
  - Run $M_i$ on input $1^n$ for $2^n/10$ steps.
    - If $M_i$ queries “$x \in B$?” for strings for which we already determined if $x \in B$ or $x \notin B$, use the same answer.
    - If $M_i$ queries “$x \in B$?” for new strings, answer that $x \notin B$.
  - Ensure that $M_i$’s answer on $1^n$ after $2^n/10$ steps is wrong.
    - If $M_i$ accepts $1^n$, for all strings $x \in \{0, 1\}^n$, let $x \notin B$.
    - If $M_i$ rejects $1^n$, take some yet unqueried $x \in \{0, 1\}^n$, and let $x \in B$.

- Each TM is represented by infinitely many $i$, and every polynomial is smaller than $2^n/10$ for large enough $n$. So no TM can decide $U_B$ in polynomial time with oracle access to $B$. 
Suppose that we have a relativizing proof that $P = NP$

Then also $P^B = NP^B$, contradicting $P^B \neq NP^B$.

Suppose that we have a relativizing proof that $P \neq NP$

Then also $P^A \neq NP^A$, contradicting $P^A = NP^A$. 
- Limits of diagonalization, relativizing results
- Oracles
- There exist $A, B \subseteq \{0, 1\}^*$ such that $P^A = NP^A$ and $P^B \neq NP^B$. 
Next time

- Space-bounded computation
- Limits on memory space