Today

- A bird’s eye overview of what we covered
- Quick intro into *parameterized complexity theory*
An overview of complexity classes

\[
\begin{align*}
P & \subseteq \text{coNP} \\
\text{NP} & \subseteq \Sigma^p_2 \subseteq \Sigma^p_3 \\
\text{PSPACE} & \subseteq \Sigma^p_2 \subseteq \Sigma^p_3 \\
\text{PH} & \subseteq \Sigma^p_2 \subseteq \Sigma^p_3 \\
\text{EXP} & \supseteq \text{PSPACE} \supseteq \text{PSPACE} \\
\end{align*}
\]
The Cook-Levin Theorem

Theorem (Cook 1971, Levin 1969)

3SAT is NP-complete.
**The Time Hierarchy Theorems**

**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 $\text{DTIME}(f(n)) \subsetneq \text{DTIME}(g(n))$.

**Theorem**

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

If $S : \mathbb{N} \to \mathbb{N}$ is a space-constructible function, then:

$$\text{DTIME}(S(n)) \subseteq \text{SPACE}(S(n)) \subseteq \text{NSPACE}(S(n)) \subseteq \text{DTIME}(2^{O(S(n))}).$$

Theorem

If $f, g : \mathbb{N} \to \mathbb{N}$ are space-constructible functions such that $f(n)$ is $o(g(n))$, then:

$$\text{SPACE}(f(n)) \subset \text{SPACE}(g(n)) \quad \text{and} \quad \text{NSPACE}(f(n)) \subset \text{NSPACE}(g(n)).$$
Theorem (Baker, Gill, Solovay 1975)

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

TQBF is PSPACE-complete.

Theorem

Let $i \geq 1$. Then $\Sigma_i \text{SAT}$ is $\Sigma_i^p$-complete and $\Pi_i \text{SAT}$ is $\Pi_i^p$-complete.
**Theorem (Karp, Lipton 1980)**

If $\text{NP} \subseteq \text{P/poly}$, then $\Sigma_2^p = \Pi_2^p$. 
Probabilistic computation

\[ P \subseteq ZPP \subseteq \text{coRP} \subseteq BPP \subseteq \text{RP} \]
Theorem (PCP)

\[ \text{NP} = \text{PCP}(\log n, 1). \]

There exists some \( \rho < 1 \) such that for all \( L \in \text{NP} \) there is a polynomial-time reduction \( R \) from \( L \) to 3SAT where for all \( x \in \{0, 1\}^* \):

- if \( x \in L \) then \( \text{val}(R(x)) = 1 \);
- if \( x \notin L \) then \( \text{val}(R(x)) < \rho \).
ETH

Definition

Let $\delta_3$ be the infimum of the set of constants $c$ for which there exists an algorithm solving 3SAT in time $O(2^{cn}) \cdot m^{O(1)}$, where $n$ is the number of variables in the $q$-SAT input and $m$ the number of clauses.

The Exponential-Time Hypothesis (ETH) states that $\delta_3 > 0$.

Theorem

The ETH implies that there is no $2^{o(n)}$-time algorithm for 3SAT and that there is no $2^{o(n+m)}$-time algorithm for 3SAT.
### Definition (distP)

$\langle L, D \rangle$ is in the class $\text{distP}$ (also called: $\text{avgP}$) if there exists a deterministic TM $M$ that decides $L$ and a constant $\epsilon > 0$ such that for all $n \in \mathbb{N}$:

$$\mathbb{E}_{x \in_R D_n} \left[ \text{time}_M(x)^\epsilon \right] \text{ is } O(n).$$
Parameterized complexity: with VC as example

- VC: given a graph $G$ and $u \in \mathbb{N}$, does $G$ have a vertex cover of size $u$?

- This problem is NP-complete, and the best algorithms that we have take exponential time in the worst case.

- This worst-case analysis takes into account every possible input.

- Can we take into account additional knowledge about the input that we might have to get more positive worst-case guarantees?
Suppose that we are dealing with an application where the value of $u$ is always much smaller than the size of the graph $G$.

Can we restrict the exponential factor in the running time to just $u$?

**Answer:** yes!
Definition

A *parameterized problem* is a language $L \subseteq \Sigma^* \times \mathbb{N}$ of pairs $(x, k)$, where $x$ is called the *main input* and $k$ is called the *parameter*.

Definition (FPT)

A parameterized problem $L \subseteq \Sigma^* \times \mathbb{N}$ is *fixed-parameter tractable* if there exist a polynomial $p$, a computable function $f$, and a deterministic TM $M$ that, when given input $(x, k)$, decides if $(x, k) \in L$ and runs in time $f(k) \cdot p(|x|)$. 
Parameterized complexity landscape


Parameterized complexity: ‘dialogues’ with your problems

- VC: NP-complete, and no $2^{o(v)}$-time algorithm (assuming ETH)

- With $u$ as parameter? Fixed-parameter tractable

- With $v - u$ as parameter? $W[1]$-complete

- With the degree $d$ of the graph as parameter? para-NP-complete

- With the treewidth $t$ of the graph as parameter? Fixed-parameter tractable

- Etc.