Recap
*What we saw last time..*

- Space-bounded computation
- Limits on memory space
- L, NL, PSPACE
- Logspace reductions
- NL-completeness
What will we do today?

- The Polynomial Hierarchy
- Bounded quantifier alternation
- Alternating Turing machines
We saw that 3COL is NP-complete, but how about the following problem?

3COL-Extension = \{ (G, V_0) | G = (V, E) is an undirected graph, V_0 \subseteq V, and each 3-coloring of the vertices in V_0 can be extended to a proper 3-coloring of the entire graph G \}

There seems to be no single (polynomial-size) certificate for yes-inputs

It is a “∀∃-type” question

We need a different complexity class to capture the complexity of 3COL-Extension
The complexity class $\Sigma^p_2$

Definition (NP)

A language $L \subseteq \{0, 1\}^*$ is in the class NP if there is a polynomial $q : \mathbb{N} \to \mathbb{N}$ and a polynomial-time Turing machine $M$ such that for every $x \in \{0, 1\}^*$:

$$x \in L \quad \text{if and only if} \quad \text{there exists some} \ u \in \{0, 1\}^{q(|x|)} \ \text{such that} \ M(x, u) = 1.$$ 

Definition (coNP)

A language $L \subseteq \{0, 1\}^*$ is in the class coNP if there is a polynomial $q : \mathbb{N} \to \mathbb{N}$ and a polynomial-time Turing machine $M$ such that for every $x \in \{0, 1\}^*$:

$$x \in L \quad \text{if and only if} \quad \text{for all} \ u \in \{0, 1\}^{q(|x|)} \ \text{it holds that} \ M(x, u) = 1.$$
The complexity class $\Sigma_2^p$

**Definition ($\Sigma_2^p$)**

A language $L \subseteq \{0, 1\}^*$ is in the class $\Sigma_2^p$ if there is a polynomial $q : \mathbb{N} \to \mathbb{N}$ and a polynomial-time Turing machine $M$ such that for every $x \in \{0, 1\}^*$:

$$x \in L \text{ if and only if } \text{ there exists } u_1 \in \{0, 1\}^{q(|x|)} \text{ such that for all } u_2 \in \{0, 1\}^{q(|x|)} \text{ it holds that } M(x, u_1, u_2) = 1.$$
The complexity class $\Pi_2^p$

**Definition ($\Pi_2^p$)**

A language $L \subseteq \{0, 1\}^*$ is in the class $\Sigma_2^p$ if there is a polynomial $q : \mathbb{N} \to \mathbb{N}$ and a polynomial-time Turing machine $M$ such that for every $x \in \{0, 1\}^*$:

$$x \in L \text{ if and only if } \text{ for all } u_1 \in \{0, 1\}^{q(|x|)} \text{ there exists } u_2 \in \{0, 1\}^{q(|x|)} \text{ such that } M(x, u_1, u_2) = 1.$$ 

- It turns out that 3COL-Extension is $\Pi_2^p$-complete.
The complexity classes $\Sigma^p_i$

<table>
<thead>
<tr>
<th>Definition ($\Sigma^p_i$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Let $i \geq 1$. A language $L \subseteq {0, 1}^<em>$ is in the class $\Sigma^p_i$ if there is a polynomial $q : \mathbb{N} \rightarrow \mathbb{N}$ and a polynomial-time Turing machine $M$ such that for every $x \in {0, 1}^</em>$:</td>
</tr>
<tr>
<td>$x \in L$ if and only if there exists $u_1 \in {0, 1}^{q(</td>
</tr>
<tr>
<td>$\vdots$</td>
</tr>
<tr>
<td>for all $u_i \in {0, 1}^{q(</td>
</tr>
<tr>
<td>it holds that $M(x, u_1, \ldots, u_i) = 1$. if $i$ is even,</td>
</tr>
<tr>
<td>$\vdots$</td>
</tr>
<tr>
<td>there exists $u_i \in {0, 1}^{q(</td>
</tr>
<tr>
<td>such that $M(x, u_1, \ldots, u_i) = 1$. if $i$ is odd.</td>
</tr>
</tbody>
</table>
The complexity classes $\Pi^p_i$

### Definition ($\Pi^p_i$)

Let $i \geq 1$. A language $L \subseteq \{0, 1\}^*$ is in the class $\Pi^p_i$ if there is a polynomial $q : \mathbb{N} \to \mathbb{N}$ and a polynomial-time Turing machine $M$ such that for every $x \in \{0, 1\}^*$:

$$x \in L \text{ if and only if } \forall u_1 \in \{0, 1\}^{q(|x|)} \exists u_2 \in \{0, 1\}^{q(|x|)} \text{ such that } \ldots$$

It holds that $M(x, u_1, \ldots, u_i) = 1$. if $i$ is odd,

$$\vdots$$

$$\forall u_i \in \{0, 1\}^{q(|x|)} \text{ there exists } u_i \in \{0, 1\}^{q(|x|)} \text{ such that } M(x, u_1, \ldots, u_i) = 1. \text{ if } i \text{ is even.}$$
The Polynomial Hierarchy (PH)

Definition ($\Sigma^p_0$, $\Pi^p_0$, PH)

$\Sigma^p_0 = \Pi^p_0 = P$

$\text{PH} = \bigcup_{i \geq 0} \Sigma^p_i.$

- Some relations:
  - $\Pi^p_i = \{ \overline{L} \mid L \in \Sigma^p_i \}$
  - $\Sigma^p_1 = \text{NP}, \Pi^p_1 = \text{coNP}$
  - $\Sigma^p_i \subseteq \Pi^p_{i+1} \subseteq \Sigma^p_{i+2}, \quad \Pi^p_i \subseteq \Sigma^p_{i+1} \subseteq \Pi^p_{i+2}$
  - $\Sigma^p_i \subseteq \Sigma^p_{i+1}, \Pi^p_i \subseteq \Pi^p_{i+1}$
  - $\Sigma^p_i \cup \Pi^p_i \subseteq \text{PSPACE}$
  - $\text{PH} \subseteq \text{PSPACE}$
“Collapse” of the hierarchy

- Statements like “P ≠ NP” and “NP ≠ coNP” are widely believed conjectures.
- We can use these as assumptions to show some results.
  - E.g., assuming that P ≠ NP, NP-complete problems are not in P.
- For some results, stronger conjectures seem necessary.
- Another conjecture: “the PH does not collapse”
  - “the PH collapses to P” \( \text{PH} = \text{P} \)
  - “the PH collapses to the \( i \)th level” \( \text{PH} = \Sigma^P_i \)

**Theorem**

Let \( i \geq 1 \). If \( \Sigma^P_i = \Pi^P_i \), then \( \text{PH} = \Sigma^P_i \).

If P = NP, then \( \text{PH} = \text{P} \).
QBF problems complete for $\Sigma^p_i$ and $\Pi^p_i$

- $\Sigma_i\text{SAT} = \{ \varphi = \exists u_1 \forall u_2 \ldots Q_i u_i \psi(u_1, \ldots, u_i) : \varphi \text{ is a true QBF} \}$, where each $u_j = (u_{j,1}, \ldots, u_{j,\ell})$ is a sequence of propositional variables, $\exists u_j$ stands for $\exists u_{j,1} \exists u_{j,2} \ldots \exists u_{j,\ell}$, and $\forall u_j$ for $\forall u_{j,1} \forall u_{j,2} \ldots \forall u_{j,\ell}$

- $\Pi_i\text{SAT} = \{ \varphi = \forall u_1 \exists u_2 \ldots Q_i u_i \psi(u_1, \ldots, u_i) : \varphi \text{ is a true QBF} \}$

**Theorem**

Let $i \geq 1$. Then $\Sigma_i\text{SAT}$ is $\Sigma^p_i$-complete and $\Pi_i\text{SAT}$ is $\Pi^p_i$-complete (both under polynomial-time reductions).
Oracle characterizations of $\Sigma_p^i$ and $\Pi_p^i$

**Theorem**

Let $i \geq 2$. Then $\Sigma_p^i = \text{NP}^{\Sigma_{i-1}^p \text{SAT}}$ and $\Pi_p^i = \text{coNP}^{\Sigma_{i-1}^p \text{SAT}}$.

- (Or replace $\Sigma_{i-1}^p \text{SAT}$ by any $\Sigma_{i-1}^p$-complete or $\Pi_{i-1}^p$-complete problem.)

- This is often written as: $\Sigma_p^i = \text{NP}^{\Sigma_{i-1}^p}$ and $\Pi_p^i = \text{coNP}^{\Sigma_{i-1}^p}$
Configuration graphs

Configurations $C$ consist of:
1. tape contents
2. tape head positions
3. state $q \in Q$

Configuration graph of a TM $M$ on some input $x$:

- Nodes are all the configurations that are reachable from the initial configuration $C_0$
- Edge from $C$ to $C'$ if applying one of the transition functions in $C$ results in $C'$
Alternating Turing machines

Definition (Alternating Turing machines; ATMs)

- Instead of a single transition function $\delta$, there are two transition functions $\delta_1, \delta_2$.
- The set $Q \setminus \{q_{\text{acc}}, q_{\text{rej}}\}$ is partitioned into $Q_{\exists}$ and $Q_{\forall}$.
- Executions of alternating TMs are defined using a labeling procedure on the configuration graph. Repeatedly apply, until a fixpoint is reached:
  - Label each configuration with $q_{\text{acc}}$ with “accept.”
  - If a configuration $c$ with $q \in Q_{\exists}$ has an edge to a configuration $c'$ that is labeled with “accept,” then label $c$ with “accept.”
  - If a configuration $c$ has a state $q \in Q_{\forall}$ and both configurations $c', c''$ that are reachable from it in the graph are labeled with “accept,” then label $c$ with “accept.”
- The TM accepts the input if the starting configuration is labeled with “accept.”
- The TM runs in time $T(n)$ if for every input $x$ and for every possible sequence of transition function choices, the machine halts after at most $T(|x|)$ steps.
Alternating Turing machines (ct’d)

$C_0 \exists \leadsto$ so the ATM accepts the input

- $\bullet$ = reject
- $\bullet$ = accept
ATIME, ΣᵢTIME, and ΠᵢTIME

**Definition (ATIME)**

Let $T : \mathbb{N} \to \mathbb{N}$ be a function. A decision problem $L \subseteq \{0, 1\}^*$ is in $\text{ATIME}(T(n))$ if there exists an ATM that decides $L$ and that runs in time $O(T(n))$.

**Definition (ΣᵢTIME)**

Let $T : \mathbb{N} \to \mathbb{N}$ be a function. A decision problem $L \subseteq \{0, 1\}^*$ is in $\Sigma_i \text{TIME}(T(n))$ if there exists an ATM that decides $L$, that runs in time $O(T(n))$, whose initial state is in $Q_{\exists}$, and that on every input and on every path in the configuration graph alternates at most $i - 1$ times between $Q_{\exists}$ and $Q_{\forall}$.

**ΠᵢTIME** is defined similarly to $\Sigma_i \text{TIME}$, with the difference that the initial state of the ATM is in $Q_{\forall}$.
Theorem

\[ \text{PSPACE} = \bigcup_{c \geq 0} \text{ATIME}(n^c). \]

Theorem

Let \( i \geq 1 \). Then:

\[ \Sigma_i^P = \bigcup_{c \geq 0} \Sigma_i \text{TIME}(n^c) \]
\[ \Pi_i^P = \bigcup_{c \geq 0} \Pi_i \text{TIME}(n^c). \]
An overview of complexity classes

$L \subseteq NL \subseteq P \\
NP \subseteq \Sigma_2^p \subseteq \Pi_2^p \subseteq \Sigma_3^p \\
coNP \subseteq \Pi_2^p \subseteq \Pi_3^p \subseteq \Sigma_3^p \\
NP \subseteq \Sigma_2^p \subseteq \Sigma_3^p \\
PH \subseteq \text{PSPACE} \subseteq \text{EXP} \\
L \subseteq NL \subseteq P \\
NP \subseteq \Sigma_2^p \subseteq \Sigma_3^p \\
coNP \subseteq \Pi_2^p \subseteq \Pi_3^p \\
NP \subseteq \Sigma_2^p \subseteq \Sigma_3^p \\
PH \subseteq \text{PSPACE} \subseteq \text{EXP}
The classes $\Sigma^p_i$ and $\Pi^p_i$

The Polynomial Hierarchy

$\Sigma^p_i$-complete and $\Pi^p_i$-complete QBF problems

Characterizations using oracles and ATMs
Next time

- A “breather”
- Time to reflect on what we’ve done so far
- Requests for things to recap?