Exercise 1.

You have seen during the course the quantifier definition of the polynomial hierarchy: \( L \in \Sigma^p_i \) if there exists a polytime TM \( M \) and a polynomial \( g \) such that:

\[
x \in L \iff \exists u_1 \in \{0, 1\}^{g(|x|)} \forall u_2 \in \{0, 1\}^{g(|x|)} \ldots Q_i u_i \in \{0, 1\}^{g(|x|)} M(x, u_1, u_2, \ldots, u_i) = 1.
\]

You have also seen an alternative definition using oracles:

\[
\begin{align*}
\Sigma^p_0 &= P \\
\Sigma^p_i &= \text{NP}^{\text{PH}}
\end{align*}
\]

With \( \text{PH} \) being defined as \( \bigcup_{i \geq 0} \Sigma^p_i \).

1. Prove the equivalence between the quantifier-based definition and the oracle-based definition.

2. Give an oracle-based definition for \( \Pi^p_i \).

3. Show that \( \Sigma^p_i \) is closed under polynomial-time many-one reduction.

4. Show that if there exist a PH-complete problem, then the polynomial hierarchy collapses.

5. Show that if \( \Sigma^p_i = \Sigma^p_{i+1} \), then \( \text{PH} = \Sigma^p_i \).

6. Propose a family of problems \( (S_i) \) such that \( S_i \) is \( \Sigma^p_i \)-complete.

Exercise 2.

Let \( \text{MIN}^{\text{DNF}} \) denote the languages of all tuples \( (\phi, k) \) such that \( \phi \) is a SAT formula in disjunctive normal form, and \( k \in \mathbb{N} \) is such that there exists an equivalent formula \( \phi' \) of encoding size \( ||\phi'|| \leq k \). More formally,

\[
\text{MIN}^{\text{DNF}} = \{(\phi, k) \in \text{DNF} \times \mathbb{N} : \exists \phi', ||\phi'|| \leq k, \forall x, \phi(x) = \phi'(x)\}.
\]

Show that \( \text{MIN}^{\text{DNF}} \in \text{NP}^{\text{NP}} \).

Exercise 3.

A circuit lower bound for the polynomial hierarchy. The aim of this exercise is to show that for each \( k \), there is a language \( L_k \in \Sigma^p_2 \) which cannot be computed by circuits of size \( O(n^k) \).

1. Show that the result is true if \( \text{NP} \not\subseteq \text{P/poly} \). Is it equivalent to \( \text{NP} \not\subseteq \text{P/poly} \)?

2. Show that if \( \text{NP} \not\subseteq \text{P/poly} \), then it is enough to prove the claim for some \( \Sigma^p_i \), \( i \geq 2 \), instead of \( \Sigma^p_2 \).

3. Show that there are \( \leq 3^k s^{2s} \) circuits of size \( s \) (inputs are in the circuit size, gates are \( \land, \lor, \neg \), and they are restricted to take at most 2 arguments).

4. Prove that, for sufficiently large \( n \), there are less circuits of size \( n^k \lfloor \log n \rfloor \) than binary words of size \( n^{k+1} \).
Let \( x^1, \ldots, x^{2^n} \) be the sequence of all words in \( \{0,1\}^n \) ordered lexicographically. Given a binary word \( a \in \{0,1\}^{n^k+1} \) of size \( n^k+1 \) we define a subset \( L_a \subset \{0,1\}^n \) as \( x^i \in L_a \iff i \leq n^k+1 \land a_i = 1 \).

**5.** Prove that if \( a, b \in \{0,1\}^{n^k+1} \) are distinct, then \( L_a \neq L_b \).

**6.** Prove that, for sufficiently large \( n \), there exists \( a \) such that \( L_a \) cannot be recognized by a circuit of size \( n^k \lceil \log n \rceil \).

**7.** Prove that \( L_a \) can be recognized by a circuit of size \( Kn^k+2 \) for some \( K > 0 \).

For every \( n \) let \( C_n \) be the set of circuits of size \( Kn^k+2 \) that recognize functions which cannot be recognized by circuits of size \( n^k \lceil \log n \rceil \). The previous parts show that \( C_n \) is nonempty for sufficiently large \( n \). For every such \( n \) let \( C_n \in C_n \) be the circuit whose binary description is the smallest in the lexicographic order of \( \{0,1\}^* \). Define \( L_k \) as \( x \in L_k \iff C_{|x|}(x) = 1 \).

**8.** Prove that \( L_k \) belongs to \( \Sigma_i^p \) for some \( i \geq 2 \) but it cannot be recognized by circuits of size \( O(n^k) \).
   (Note: \( i = 4 \) is enough.)

**9.** Conclude.

**Exercise 4.**

A language is said to be **tally** (or unary), if it is included in a unary alphabet \( \{a\}^* \) for a fixed symbol \( a \).

**1.** Show that every tally language belongs to \( P/1 \).

**2.** Prove that every language \( L \subset \{0,1\}^* \) belongs to \( P/2^n \).

**3.** Show that there exist undecidable tally languages.

**4.** Show that there exists a **decidable** tally language that does not belong to \( P \).

**5.** Show that if there exists an \( NP \)-hard tally language, then \( P = NP \) (Berman’s theorem).

**Exercise 5.**

Prove that \( NP \subseteq P/\log \) implies \( P = NP \).