Definition 1 \((\text{NP/poly})\). A decision problem \(L \subseteq \Sigma^*\) is in the complexity class \(\text{NP/poly}\) if there exists:

- polynomials \(p, q : \mathbb{N} \rightarrow \mathbb{N}\);
- a polynomial-time Turing machine \(M\) (the verifier); and
- a sequence \(\{\alpha_n\}_{n \in \mathbb{N}}\) with \(\alpha_n \in \{0, 1\}^{q(n)}\) for each \(n \in \mathbb{N}\) (a family of advice strings)

such that for every \(x \in \Sigma^*\):

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

One can equivalently define \(\text{NP/poly}\) as the class of all decision problems decidable by a polynomial-time nondeterministic Turing machine that has access to a polynomial-length family of advice strings. (You may use either definition (or both) in your solutions.)

Definition 2 \((\text{coNP/poly})\).

\[\text{coNP/poly} = \{ L \subseteq \Sigma^* \mid \exists L = (\Sigma^* \setminus L) \in \text{NP/poly} \}.\]

Definition 3 \((\Sigma_p^i/\text{poly} \text{ and } \Pi_p^i/\text{poly})\). The complexity classes \(\Sigma_p^i/\text{poly}\) and \(\Pi_p^i/\text{poly}\), for \(i \geq 2\), are defined analogously—by taking the definitions of \(\Sigma_p^2/\text{poly}\) and \(\Pi_p^2/\text{poly}\) and adding a polynomial-size family of advice strings that the verifier machine is given access to.

Question 1 \((3pts; a: 2pts, b: 1pt)\). In this question, you will prove that if \(\text{NP} \subseteq \text{coNP/poly}\), then the Polynomial Hierarchy collapses. The general proof line will be as follows.

You will show that if \(\text{NP} \subseteq \text{coNP/poly}\), then \(\Sigma_p^3 \subseteq \text{NP/poly}\). The following (true) statement, which you do not have to prove, can then be used to show that the Polynomial Hierarchy collapses.

If \(\Sigma_p^3 \subseteq \text{NP/poly}\), then \(\Sigma_p^3 = \Pi_p^3\).

Complete the proof by doing the following.

(a) Prove that if \(\text{NP} \subseteq \text{coNP/poly}\), then \(\Sigma_p^2/\text{poly} \subseteq \text{NP/poly}\).

- \text{Hint}: as an intermediate step, show that if \(\text{NP} \subseteq \text{coNP/poly}\), then \(\Sigma_p^2 \subseteq \text{NP/poly}\).

(b) Prove that if \(\text{NP} \subseteq \text{coNP/poly}\), then \(\Sigma_p^3 \subseteq \text{NP/poly}\).

- \text{Hint}: use the statement that you proved for (a).

Definition 4. Consider the following problem 

\text{clause entailment}:

\[
\text{Input:} \text{ A propositional formula } \varphi, \text{ and a propositional clause } c \text{ (i.e., a disjunction of literals).}
\]

\[
\text{Question: } \varphi \models c? \text{ I.e., is it the case that all truth assignments } \alpha \text{ that make } \varphi \text{ true also make } c \text{ true?}
\]

Definition 5. We say that there is a \(\text{hint system}\) for \text{clause entailment} if there exists a computable function \(f : \Sigma^* \rightarrow \Sigma^*\) (called the \(\text{hint function}\)) and a polynomial-time decidable problem \(Q\) such that for each input \((\varphi, c)\) of \text{clause entailment} it holds that \(\varphi \models c\) if and only if \((\varphi, f(\varphi), c) \in Q\).

Definition 6. Let \(f : \Sigma^* \rightarrow \Sigma^*\) be a function. We say that \(f\) is \(\text{polynomial-size}\) if there exists a polynomial \(p\) such that for all \(x \in \Sigma^*\) it holds that \(|f(x)| \leq p(|x|)|\).
Question 2 (4pts: a: 1pt, b: 1pt; c: 2pts).

(a) Prove that the function \( f_0 \) that for any propositional formula \( \varphi \) outputs its truth table (over the variables appearing in \( \varphi \)) leads to a hint system for \textsc{clause entailment}. That is, if \( f_0(\varphi) \) consists of a list mentioning for each truth assignment \( \alpha : \text{Var}(\varphi) \rightarrow \{0, 1\} \) whether or not \( \alpha \) makes \( \varphi \) true.

In particular, identify a polynomial-time decidable problem \( Q \) such that for each formula \( \varphi \) and each clause \( c \) it holds that \( \varphi \models c \) if and only if \( (\varphi, f_0(\varphi), c) \in Q \). Make sure to prove that \( Q \) is polynomial-time decidable. You do not have to prove that \( f_0 \) is computable.

- Note: \( f_0 \) is not a polynomial-size function.

(b) Prove that if there is a hint system for \textsc{clause entailment} with a polynomial-time computable hint function \( f \), then \( P = \text{NP} \).

(c) Prove that if there is a hint system for \textsc{clause entailment} with a polynomial-size hint function \( f \), then the Polynomial Hierarchy collapses.

- Hint: use the fact that \( \text{NP} \subseteq \text{coNP/poly} \) implies that \( \text{PH} = \Sigma_p^P \). (This statement you proved in Question 1.)

- Hint: for different values of \( \ell \in \mathbb{N} \), consider the formula:

\[
\varphi_\ell = \bigwedge_{1 \leq i \leq (2\ell)^3} (y_i \rightarrow c_i),
\]

where \( c_1, \ldots, c_{(2\ell)^3} \) is an enumeration of all possible clauses of size 3 over the variables \( x_1, \ldots, x_\ell \).

Definition 7. Consider the following problem \textsc{set packing}:

\textit{Input:} A finite set \( U \), a set \( S \subseteq \mathcal{P}(U) \) of subsets of \( U \), and a positive integer \( k \in \mathbb{N} \) (given in unary).

\textit{Question:} Are there \( k \) sets \( S_1, \ldots, S_k \) in \( S \) that are pairwise-disjoint—that is, such that \( S_i \cap S_j = \emptyset \) for all \( 1 \leq i < j \leq k \)?

Question 3 (3pts; a: 1/2pt, b: 1/2pts, c: 1pt).

(a) Prove that there exists an algorithm that solves \textsc{set packing} in time \( n^{O(k)} \), where \( n \) denotes the size of the input.

(b) Prove that there does not exist an algorithm that solves \textsc{set packing} in time \( 2^k \cdot n^{o(k)} \), assuming the ETH.

- Hint: Consider the following reduction from \textsc{3col} to \textsc{set packing}. Let \( G = (V, E) \) be an instance of \textsc{3col} with \( |V| = n \). The reduction partitions the nodes (arbitrarily) into \( \log n \) groups \( V_1, \ldots, V_{\log n} \) consisting of at most \( n/\log n \) nodes.

It then constructs an instance \((U, S, k)\) of \textsc{set packing} as follows. We set \( k = \log n \). The set \( C \) consists of all 3-colorings \( \mu \) that are defined on exactly one of \( V_1, \ldots, V_{\log n} \) and that do not assign any two nodes connected by an edge in \( E \) to the same color. That is, \( C = \bigcup_{1 \leq i \leq \log n} C_i \) where \( C_i \) is the set of all colorings \( \mu : V_i \rightarrow \{1, 2, 3\} \) such that for no edge \( \{u, v\} \in E \) it holds that \( u \in V_i, v \in V_i \) and \( \mu(u) = \mu(v) \). The set \( U \) consists of all size-2 sets \( \{\mu_1, \mu_2\} \subseteq C \) of colorings in \( C \).

Finally, the set \( S \) is constructed as follows. For each \( \mu \in C \), we construct a set \( S_\mu \subseteq U \) as follows. Remember that each element of \( C \) corresponds to some 3-coloring of a subset of \( V \). Take an arbitrary \( \mu \in C \). Then \( S_\mu \) contains \( \{\mu, \mu'\} \) for all colorings \( \mu' \neq \mu \) such that either (1) \( \mu \) and \( \mu' \) are defined on the same set of nodes, or (2) \( \mu \) and \( \mu' \) are defined on different sets of nodes and there is some edge \( \{u, v\} \in E \) such that \( \mu \) and \( \mu' \) combined assign the same color to both \( u \) and \( v \).

- Note: you still have to prove that this reduction is correct.

(c) Prove that there does not exist an algorithm that solves \textsc{set packing} in time \( 2^{2^k} \cdot n^{o(k)} \), assuming the ETH.

- Note: for (c), it suffices to indicate where (and how exactly) your solution for (b) needs to be adapted.