Exercise 1: Closure under morphisms
Given a finite alphabet $\Sigma$, a function $f : \Sigma^* \rightarrow \Sigma^*$ is a morphism if $f(\Sigma) \subseteq \Sigma$ and for all $a = a_1 \cdots a_n \in \Sigma^*$, $f(a) = f(a_1) \cdots f(a_n)$ ($f$ is uniquely determined by the value it takes on $\Sigma$).

1. Show that $\text{NP}$ is closed under morphisms, that is: for any language $L \in \text{NP}$, and any morphism $f$ on the alphabet of $L$, $f(L) \in \text{NP}$.

2. Show that if $\text{P}$ is closed under morphisms, then $\text{P} = \text{NP}$.

Exercise 2: Unary Languages
Prove that if a unary language is $\text{NP}$-complete, then $\text{P} = \text{NP}$.

Hint: consider a reduction from SAT to this unary language and exhibit a polynomial time recursive algorithm for SAT

Exercise 3: Complete problems for levels of PH
Prove that the following problem $\Sigma_k^{\text{QBF}}$ is $\Sigma_k^P$-complete (under polynomial time reductions).

- INPUT: A quantified boolean formula $\exists X_1 \forall X_2 \exists \ldots Q_k X_k \phi$, where $X_1, \ldots X_k$ are $k$ disjoint sets of variables, $Q_k$ is the quantifier $\forall$ if $k$ is even, and the quantifier $\exists$ if $k$ is odd, $\phi$ is a boolean formula over variables $\bigcup_{i=1..k} X_i$;

- QUESTION: is the input formula true?

Define a similar problem $\Pi_k^{\text{QBF}}$ such that $\Pi_k^{\text{QBF}}$ is $\Pi_k^P$-complete.

Exercise 4: Collapse of PH

1. Show that if $\Sigma_k^P = \Pi_k^P$ for some $k$ then $\text{PH} = \Sigma_k^P$.

2. Show that if $\text{P} = \text{NP}$ then $\text{P} = \text{PH}$.

3. Prove that if $\Sigma_k^P = \Sigma_k^P + 1$ for some $k \geq 0$ then $\text{PH} = \Sigma_k^P$.

4. Show that if $\text{PH} = \text{PSPACE}$ then $\text{PH}$ collapses.

5. Do you think there is a polynomial time procedure to convert any QBF formula into a QBF formula with at most 10 variables?
Exercise 5: Oracle machines

Let \( O \) be a language. A Turing machine with oracle \( O \) is a Turing machine with a special additional read/write tape, called the oracle tape, and three special states: \( q_{\text{query}}, q_{\text{yes}}, q_{\text{no}} \). Whenever the machine enters the state \( q_{\text{query}} \), with some word \( w \) written on the oracle tape, it moves in one step to the state \( q_{\text{yes}} \) or \( q_{\text{no}} \) depending on whether \( w \in O \).

We denote by \( P^O \) (resp. \( NP^O \)) the class of languages decided in polynomial time by a deterministic (resp. non-deterministic) Turing machine with Oracle \( O \). Given a complexity class \( C \), we define \( P^C = \bigcup_{O \in C} P^O \) (and similarly for \( NP \)).

1. Prove that for any \( C \)-complete language \( L \) (for polynomial time reductions), \( P^C = P^L \) and \( NP^C = NP^L \).
2. Show that for any language \( L \), \( P^L = P^\emptyset \) and \( NP^L = NP^\emptyset \).
3. Prove that if \( NP = p^{SAT} \) then \( NP = \text{coNP} \).
4. Prove that \( \Sigma_{k+1}^P = NP^{\Sigma_k^P} \). Give an oracle characterization of \( \Pi_k^P \).
5. Deduce a simpler proof that if \( \Sigma_k^P = \Sigma_{k+1}^P \) for some \( k \), then \( PH \) collapses.

This third view of the polynomial hierarchy gives access to the class of languages \( \Delta_{k+1}^P = P^{\Sigma_k^P} \), with \( \Delta_0^P = P \).

Exercise 6: Relativization

Show that there is an oracle \( O \) such that \( P^O = NP^O \).

Exercise 7: Sort your problems

Give upper complexity bounds for the following problems:

1. **MIN-FORMULA**
   - **INPUT:** a propositional formula \( \phi \)
   - **QUESTION:** is \( \phi \) minimal, in the sense that there exists no smaller formula equivalent to \( \phi \) ?

2. **MAX-CLIQUE**
   - **INPUT:** a graph \( G \) and a natural number \( k \)
   - **QUESTION:** \( k \) is the exact size of a maximal clique in \( G \)

3. **USAT**
   - **INPUT:** a boolean formula \( \phi \)
   - **QUESTION:** is \( \phi \) satisfiable by only one assignment

Exercise 8: Which one is lying ?

Suppose that a Turing machine has access to two oracles \( A \) and \( B \), one of which is an oracle for \( QBF \), but you don’t know which. Show that \( QBF \) can still be decided in polynomial time by such a machine.
Exercise 9: Bounded number of queries to the oracle

A \( k \)-query oracle Turing machine is an oracle Turing machine that can access the \( q_{\text{query}} \) state at most \( k \) times. Given an oracle \( O \), define \( P^{O,k} \) the class of languages that can be decided in deterministic-polynomial time by a \( k \)-query Turing machine with oracle \( O \).

1. Show that \( \text{NP} \cup \text{coNP} \subseteq P^{SAT,1} \).
2. Assuming \( \text{NP} \neq \text{coNP} \), prove that the first inclusion above is strict.

Exercise 10: The Difference Hierarchy

Let \( \text{DP} \) be the class of languages of the form \( L_1 \cap L_2 \), where \( L_1 \in \text{NP} \) and \( L_2 \in \text{coNP} \). (In other words a language in \( \text{DP} \) is the difference of two \( \text{NP} \) languages.)

We consider the problem **EXACT-INDSET**:

- **INPUT:** a graph \( G \) and an integer \( k \)
- **OUTPUT:** does the maximum size of an independent set of \( G \) is \( k \)? That is whether \( G \) has an independent set of size \( k \), and all other independent sets of \( G \) have size at most \( k \). Recall that an independent set of a graph is a set \( I \) of vertices such that no two vertices of \( I \) are connected by an edge.

1. Show that \( P^{SAT,1} \subseteq \text{DP} \subseteq P^{SAT,2} \).
2. Prove that **EXACT-INDSET**:
   (a) is in \( \Sigma_2^P \cap \Pi_2^P \);
   (b) is in \( \text{DP} \);
   (c) is \( \text{DP} \)-complete (under polynomial time reductions).