Reminder. Homework must be typeset with \LaTeX preferred. Make sure you understand the course collaboration and honesty policy before beginning this assignment. Collaboration is permitted, but you must write the solutions by yourself without assistance. You must also identify your collaborators. Assignments missing a collaboration statement will not be accepted. Getting solutions from outside sources such as the Web or students not enrolled in the class is strictly forbidden.

Problem 1 (Hierarchy theorems and padding). Recall the “padding argument” proof that complexity collapses such as $\text{P} = \text{NP}$ would scale up to $\text{EXP} = \text{NEXP}$. (Or equivalently, the complexity separation $\text{EXP} \neq \text{NEXP}$ would scale down to $\text{P} \neq \text{NP}$.)

(a) Show that if $\text{NTIME}(n) \subseteq \text{DTIME}(n^2)$, then $\text{NTIME}(n^2) \subseteq \text{DTIME}(n^4)$.

(b) Show that for every $k \geq 1$, we have $\text{P} \neq \text{NTIME}(n^k)$.

Hint: Part (a) is not directly useful here, but the idea behind it is.

Problem 2 (NP vs. coNP relative to an oracle). The Baker-Gill-Solovay Theorem (Theorem 3.7 in Arora-Barak) shows that there is an oracle $A$ relative to which $\text{P}^A \neq \text{NP}^A$. This problem will walk you through a proof of the stronger result that $\text{NP}^A \neq \text{coNP}^A$ for some oracle. (The writeup is long, but the parts you have to fill in should be short!)

(a) A DNF formula $D$ is an OR of “terms,” where each term is an AND of literals. The width of a DNF is the maximum number of literals appearing in any term and the size is the number of terms. For example, $(x_1 \land \overline{x_2}) \lor (x_2 \land x_3 \land x_4)$ is a DNF of width 3 and size 2.

Believe it or not, the whole proof rests on the following simple combinatorial fact\(^1\): The function $\text{AND}_N(x_1, \ldots, x_N) = x_1 \land \cdots \land x_N$ cannot be computed by a DNF $D(x_1, \ldots, x_N)$ of width $< N$. Prove this fact.

(b) Another way to think of an oracle $A$ is as an infinitely long vector, indexed by binary strings. That is, for $z \in \{0, 1\}^*$, let $A_z = 1$ if $z \in A$ and $A_z = 0$ if $z \notin A$. Recall that to query $A$, a TM can write a string $z$ to its oracle tape and receive the bit $A_z$.

Let $f = \{f_n : \{0, 1\}^{2^n} \to \{0, 1\}\}$ be a sequence of Boolean functions. Define the unary language\(^2\)

$$L_f(A) = \{1^n \mid f_n(A) = 1\}.$$
Let $f_n(A) = \text{AND}_{z \in \{0,1\}^n}(A_z)$, so that $L_f(A)$ consists of the strings $1^n$ for which $z \in A$ for all $z \in \{0,1\}^n$. Show that $L_f(A) \in \text{coNP}$ for every oracle $A$.

(c) Now our job is construct an oracle $A$ for which $L_f(A) \notin \text{NP}$. We’ll do this by first arguing that the output of an $\text{NP}$ machine is just a DNF applied to the oracle $A$.

Let $M$ be a nondeterministic oracle Turing machine running in time $T(n)$. Show that for every $n \in \mathbb{N}$ there exists a DNF formula $D_n$ of width at most $T(n)$ and size at most $2T(n)$ such that for every oracle $A$,

$$M^A(1^n) = 1 \iff D_n(A) = 1,$$

again regarding $A$ as an infinite vector $(A_z)_{z \in \{0,1\}^*}$.

Hint 1: When run on an input of length $n$, the machine $M$ can query the oracle at most $T(n)$ times.

Hint 2: A binary tree of depth $T$ has at most $2^T$ leaves.

(d) Next we diagonalize against all nondeterministic machines running in time $T(n) = 2^n/10$ to instantiate the oracle $A$. Let $M_1, M_2, \ldots$ be an enumeration of such machines, with each machine appearing infinitely often in the list. We construct an oracle $A^*$ (details below) such that for every machine $M_i$ in the enumeration, there exists an input $1^{n_i}$ such that $M_i^{A^*}(1^{n_i}) = 1 \iff 1^{n_i} \notin L_f(A^*)$.

Using this guarantee of the oracle $A^*$, put everything together to conclude that $\text{NP}^{A^*} \neq \text{coNP}^{A^*}$.

Hint: Details for how to construct $A^*$ are provided below for your interest and enjoyment, but you do not need to understand anything about how $A^*$ works to solve the problem.

We construct $A^*$ iteratively as follows. We initialize $A^*$ to be the empty language, and in each round $i$, commit to including or excluding strings of a certain length $n_i$ in $A^*$. We choose $n_i$ large enough so that no string of length $n_i$ has had its fate decided in any previous round.

Let $D_{n_i}$ be the DNF formula guaranteed by part (c) capturing the behavior of machine $M_i$ on input $1^{n_i}$. By part (a), there exists an oracle $A^{(i)}$ such that

$$D_{n_i}(A^{(i)}) \neq f_{n_i}(A^{(i)}).$$

Actually, something stronger is true. Since $f_{n_i}(A)$ only depends on the values of $A_z$ for $z \in \{0,1\}^{n_i}$, we may assume that $A_z^{(i)} = A^*_z$ for every $|z| < n_i$.

Now for all of the (finitely many) $z$ for which the variable $A_z$ appears in the DNF $D_{n_i}$, set $A_z^* = A_z^{(i)}$. That is, for each such $z$, commit to including $z$ in $A^*$ if $z \in A^{(i)}$ and commit to excluding $z$ from $A^*$ if $z \notin A^{(i)}$. Then by part (b),

$$M_i^{A^*}(1^{n_i}) = 1 \iff D_{n_i}(A^*) = 1 \iff f_{n_i}(A^*) = 0 \iff 1^{n_i} \notin L_f(A^*).$$
Problem 3 (*Bonus Problem*, NP vs. coNP relative to a random oracle). Are oracles that separate NP from coNP rare, or are they common? In this problem, you’ll show that NP\(^A \neq \coNP^A\) not only for some contrived oracle \(A\), but for almost all oracles. This whole problem is all just for fun and you can solve any of the subparts independently assuming the previous ones.

(a) For \(N \in \mathbb{N}\), let \(N = sw\) where \(w = \Theta(\log N)\) and \(s = \Theta(N/\log N)\) are chosen so that 
\[
(1 - 2^{-w})^s = \frac{1}{2} \pm o(1). 
\]
Define the function \(C_N(x_{1,1}, \ldots, x_{s,w}) = \bigwedge_{i=1}^s \bigvee_{j=1}^w x_{i,j}\).

If we take \(f_n = C_{2^n}\), show that \(L_{f_n}(A) \in \coNP^A\) for every oracle \(A\).

(b) Show that there is a constant \(c\) such that for every DNF \(D\) of width at most \(cN/\log N\),
\[
\Pr_{x \sim \{0,1\}^N}[D(x) \neq C_N(x)] \geq 0.1.
\]

(c) Let \(A \subseteq \{0,1\}^*\) be a random oracle. That is, for every string \(z \in \{0,1\}^*\), include \(z\) in \(A\) independently with probability 1/2. Show that \(\NP^A \neq \coNP^A\) with probability at least 0.99 over the choice of the random oracle \(A\).

\(3\) Complexity class separations satisfy a zero-one law, which implies that the probability of a separation is actually 1.