

CS 535: Complexity Theory, Fall 2026

Homework 4

Due: 11:59PM, Friday, October 2, 2026.

Reminder. Homework must be typeset with $\text{\LaTeX}$ preferred. Problem 1 (all parts) and Problem 3 are required. Problem 2 is optional.

Problem 1 (*Required* NP vs. PSPACE 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 that there is also an oracle A relative to which $\text{NP}^A \neq \text{PSPACE}^A$. This proof will hopefully illuminate a general recipe for proving oracle separations.

- (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, $(x_1 \wedge \overline{x_2}) \vee (x_2 \wedge x_3 \wedge x_4)$ is a DNF of width 3 and size 2.

Believe it or not, the whole proof rests on a simple combinatorial fact¹: *Prove that the function $\text{XOR}_N(x_1, \dots, x_N) = x_1 \oplus \dots \oplus x_N$ cannot be computed by a DNF $D(x_1, \dots, x_N)$ of width $< N$.*

- (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 .

For an oracle A , define the unary language

$$\begin{aligned} L_{\text{XOR}}(A) &= \{1^n \mid \text{XOR}_{2^n}((A_z)_{z \in \{0,1\}^n}) = 1\} \\ &= \{1^n \mid z \in A \text{ for an odd number of strings } z \in \{0,1\}^n\}. \end{aligned}$$

Show that $L_{\text{XOR}}(A) \in \text{PSPACE}^A$ for every oracle A .

- (c) Now our job is construct an oracle A for which $L_{\text{XOR}}(A) \notin \text{NP}^A$. We'll do this by first arguing that the output of an NP^A 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 $2^{O(T(n))}$ such that for every oracle A ,

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

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

¹Well, in the same way that Baker-Gill-Solovay rests on the fact that OR_N cannot be computed by a “decision tree” of depth $< N$.

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.

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

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 $\geq n_i$ has had its fate decided in any previous round.

Let D_{n_i} be the DNF formula of width $2^n/10$ guaranteed by part (c) capturing the behavior of machine M_i on input 1^{n_i} .

- (d) Use part (a) to show that there exists an oracle $A^{(i)}$ agreeing with every previous commitment such that

$$D_{n_i}(A^{(i)}) \neq \text{XOR}_{2^{n_i}}((A_z^{(i)})_{z \in \{0,1\}^{n_i}}).$$

Moreover, explain why this disagreement will continue to hold under every subsequent assignment to the remaining oracle bits if we permanently fix both:

- every bit A_z appearing in D_{n_i} , and
- every bit A_z with $|z| = n_i$

to their values in $A^{(i)}$.

(Your job is now done.)

Using the oracle constructed in part (d), for every string z such that A_z appears in D_{n_i} or $|z| = n_i$, commit to including z in A^* if $z \in A^{(i)}$, and commit to excluding z from A^* otherwise. These commitments are consistent with all previous commitments, and only finitely many strings are committed in each round. After carrying out all rounds, exclude from A^* all strings whose membership was never determined. Now by parts (c) and (d), for every i we have

$$M_i^{A^*}(1^{n_i}) = 1 \iff D_{n_i}(A^*) = 1 \iff \text{XOR}_{2^{n_i}}((A_z^*)_{z \in \{0,1\}^{n_i}}) = 0 \iff 1^{n_i} \notin L_{\text{XOR}}(A^*).$$

Let M be any nondeterministic TM running in time $T_M(n) = \text{poly}(n)$. Let n_* be sufficiently large so that $T_M(n_*) \leq 2^{n^*}/10$. Since M appears infinitely often in our enumeration, there exists some $n \geq n_*$ such that $M^{A^*}(1^n) = 1 \iff 1^n \notin L_{\text{XOR}}(A^*)$. Hence, $L_{\text{XOR}}(A^*)$ is not decided by any poly-time NTM, so $L_{\text{XOR}}(A^*) \notin \text{NP}^{A^*}$.

On the other hand, we showed in part (b) that $L_{\text{XOR}}(A) \in \text{PSPACE}^A$ for every oracle A , and in particular for A^* . Hence $\text{NP}^{A^*} \neq \text{PSPACE}^{A^*}$.

Problem 2 (*Optional*, NP vs. PSPACE relative to a random oracle). Are oracles that separate NP from PSPACE rare, or are they common? In this problem, you'll show that $\text{NP}^A \neq \text{PSPACE}^A$ not only for some contrived oracle A , but for *almost all* oracles.

- (a) Show that for sufficiently large N , for every DNF D of width at most $\sqrt{N}$,

$$\Pr_{x \sim \{0,1\}^N} [D(x) \neq \text{XOR}_N(x)] \geq 0.1.$$

- (b) 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 $\text{NP}^A \neq \text{PSPACE}^A$ with probability at least 0.99 over the choice of the random oracle A .²

Problem 3 (*Required* Generalized Savitch's Theorem). Let $S(n)$ be space-constructible and $T(n)$ be time-constructible functions such that $S(n) \leq T(n) \leq 2^{O(S(n))}$. Show that if a language L is decidable by a nondeterministic TM running in both time $T(n)$ and space $S(n)$ (that is, on every input x and every sequence of nondeterministic choices, the NTM takes at most $T(|x|)$ steps and uses at most $S(|x|)$ work tape cells), then $L \in \text{SPACE}(S(n) \log T(n))$.

²Complexity class separations satisfy a *zero-one law*, which implies that the probability of a separation is actually 1.