CS 278: Computational Complexity Theory
Homework 4

Due: **Sunday, December 14, 2025, 11:59pm Pacific Time**

Fall 2025

Instructions:

- Collaboration is allowed, but you must write up your solutions *by yourself* and in your own words. List all collaborators and any external resources you used.
- Write your solutions in LaTeX and submit a single PDF to Gradescope under "HW4".
- **Deadline:** 11:59pm Pacific Time on **Sunday, December 14, 2025**.
- Late submissions lose **10%** per day (e.g., three days late $\to 0.9^3$ of your score).
- The maximum *raw* score of this homework is 160. There are 2 problems, each worth 80 points, with many subparts. However, you only need 100 points to get full credit.
- Let $n = \min\{\text{your raw score on this homework}, 100\}$. The contribution of this homework to your course grade is
$$a_4 = \frac{n}{100} \cdot 12.5.$$
- Let a_1, a_2, a_3, a_4 be the contributions from Homeworks 1–4. Your final homework component is
$$\min(a_1 + a_2 + a_3 + a_4, 50).$$
- In other words, you do *not* need to solve every problem on every homework to get full homework credit.

1 Problem 1 (80 pts): Near-linear MA protocol for Counting Orthogonal Vectors

The *Counting Orthogonal Vectors (#OV)* problem is defined as follows: given two sets of vectors $A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_n\}$ where each $a_i, b_i \in \{0,1\}^d$, count the number of pairs (i, j) such that $\langle a_i, b_j \rangle = 0$. Here the inner product is over the integers (or reals), i.e., $\sum_{k=1}^{d} a_{i,k} b_{j,k} = 0$. The "standard" algorithm takes $O(n^2 d)$ time. In this problem, we will show a near-linear-time MA protocol for #OV. Let p be a prime number such that $p > n$.

(a) **(20 pts) Polynomial Formulation.**
Let $\mathbb{F}_p$ be a prime field with $p > n$. For a fixed vector $v \in \{0,1\}^d$, define a multivariate polynomial $P_v(x_1, \ldots, x_d)$ over $\mathbb{F}_p$ such that for any vector $u \in \{0,1\}^d$:

$$P_v(u) = 1 \iff \langle u, v \rangle = 0$$

and

$$P_v(u) = 0 \iff \langle u, v \rangle \neq 0.$$

Ideally, the degree of P_v should be relatively low (e.g., degree d or close to it).

Hint: write $\langle u, v \rangle$ as a simple AC^0 circuit and arithmetize it.

(b) **(15 pts) Batch Verification via Polynomials.**

Building on the ideas from part (a), for the given set $A = \{a_1, \ldots, a_n\}$, define a polynomial $P_A(x_1, \ldots, x_d)$ such that for any vector $u \in \{0,1\}^d$: we have $P_A(u)$ is the number of vectors $a_i \in A$ such that $\langle a_i, u \rangle = 0$.

Furthemore, show that $P_A(x_1, \ldots, x_d)$ can be computed in time $\tilde{O}(n \cdot \mathrm{poly}(d))$.

(c) **(25 pts) The Protocol.**

Show that you can construct another univariate polynomial $Q_A(x)$ such that $Q_A(i)$ equals $P_A(b_i)$ for all $i \in [n]$, and $Q_A(x)$ can be computed in $\tilde{O}(n \cdot \mathrm{poly}(d))$ time.

Hint: You can use the fact that given a univariate polynomial $P(x)$ of degree T and n inputs $x_1, \ldots, x_n$, you can compute $P(x_1), \ldots, P(x_n)$ in time $\tilde{O}(n + T)$.

Show that given the correct $Q_A(u)$ you can solve the #OV problem in time $\tilde{O}(n \cdot \mathrm{poly}(d))$.

(d) **(25 pts) Soundness and Completeness.**

Design an MA protocol such that if Merlin sends Arthur the correct $Q_A(u)$, then Arthur accepts with probability 1; and otherwise Arthur rejects with probability at least $1 - 1/\mathrm{poly}(n)$. Show how this protocol can be used to solve the #OV problem in time $\tilde{O}(n \cdot \mathrm{poly}(d))$.

Hint: think about the IP=PSPACE protocol covered in class.

2 Problem 2 (80 pts): Complexity of Transformers

This problem asks you to reason about very simple complexity-theoretic idealizations of Transformer-style sequence models and Chain-of-Thought (CoT) reasoning. You do *not* need to know anything about practical Transformers beyond what is stated here.

Basic notions.

- A **token** is just a symbol from some finite alphabet Σ (you can think of $\Sigma = \{0, 1\}$ in this problem).
- An **autoregressive (AR) generative model** for length-T outputs is a probabilistic algorithm G_{AR} which, on input x and random coins r, produces tokens $y_1, \ldots, y_T$ one-by-one. Formally, in round t it outputs y_t as a (randomized) function of $(x, y_1, \ldots, y_{t-1}, r)$, and the total running time over all T rounds is $\mathrm{poly}(|x|, T)$.
- AC^0 is the class of constant-depth, polynomial-size Boolean circuits with unbounded fan-in AND, OR, and NOT gates. TC^0 is defined similarly, but also allows unbounded fan-in *majority* (threshold) gates.
- A **one-way function (OWF)** is a function $f : \{0, 1\}^n \to \{0, 1\}^n$ that is computable in time $\mathrm{poly}(n)$ but is hard to invert on a random input: let $x \in \{0, 1\}^n$ be a random input, and $y = f(x)$. Then for any $\mathrm{poly}(n)$-time adversary A, the probability that $A(y) = x'$ such that $f(x') = y$ is negligible.
- A **Chain-of-Thought (CoT) decoder** is an AR model that, on input x, first emits a sequence of intermediate "reasoning" tokens $z_1, z_2, \ldots, z_L$ (the *chain-of-thought*), and then emits a final answer token y. "Thinking in dots" refers to the special case where all the z_i are the same filler token "$\cdot$".

In all parts below, high-level arguments and sketches are fine; you do *not* need to give fully formal reductions, but you should clearly indicate what assumptions you are using and why they are relevant.

(a) **(20 pts) Limitations of AR models under OWF.** Assume one-way functions exist. Prove that there exists a distribution D over $\{0, 1\}^n$ that cannot be generated by any polynomial-time AR model, but can be generated by a polynomial-time algorithm that is not AR.

(b) **(25 pts) "Thinking in dots" with a TC^0 decoder.** Assume that a constant-depth decoder is in TC^0, show that the thinking-in-dots decoder is in TC^0. Here, the thinking-in-dots decoder works by in every decoding step it either emits a filler token "$\cdot$" or the answer token and halts, and it is guaranteed that it thinks in dots for at most $\mathrm{poly}(n)$ steps.

(c) **(35 pts) Why CoT does not give AC^0 decoders Parity.**
Now suppose each decoding step (including the final answer step) is computed by a uniform AC^0 circuit on the current state. As above, on input $x \in \{0,1\}^n$, the decoder first produces a chain-of-thought $z_1, \dots, z_L$ of length $L = n^{0.99}$, then a final answer y. Assume the decoding is deterministic (e.g., greedy decoding).

You may use the following standard average-case hardness theorem for Parity:

> *Theorem (Parity vs. AC^0, informal).* For every constant depth d, for every $\delta \in (0,1)$, it holds for all sufficiently large n, every depth-d AC^0 circuit C of polynomial size satisfies
>
> $$\Pr_{x \leftarrow \{0,1\}^n}[C(x) = \mathrm{Parity}(x)] \le \tfrac{1}{2} + 2^{-\Omega(n^\delta)}.$$

Give a careful but high-level argument that even with a chain-of-thought of length $L = n^{0.99}$, such an AC^0-based decoder still cannot compute the Parity function with high success probability on a random input x.

(Optional) Solution placeholders
You may use the following headings for your writeup; remove them if not needed.

Solution to Problem 1

Part (a): Polynomial Formulation

Part (b): Batch Verification via Univariate Polynomials

Part (c): The Protocol

Part (d): Soundness and Completeness

Solution to Problem 2

Part (a): Limitations of AR models under OWF

Part (b): "Thinking in dots" with a $\mathbf{TC}^0$ decoder

Part (c): Why CoT does not give $\mathbf{AC}^0$ decoders Parity