COMP 590: Computational Complexity **Spring 2026**

Problem Set 4

Instructions: You **must** typeset your solution in LaTeX using the provided template. Please submit your problem set via Canvas. Include your name and the names of any collaborators at the top of your submission.

Name:
Collaborators:

Acknowledgment: Most problems come from the Arora Barak textbook and are reproduced here for convenience and typo fixes.

Problem 1: The IP Definition.

(a) Let $\mathbf{IP}'$ denote the class obtained by requiring in the completeness condition for **IP** that there exists a single prover P for every $x \in L$ (rather than requiring that for every $x \in L$ there exists a prover). Prove that $\mathbf{IP}' = \mathbf{IP}$.

(b) Let $\mathbf{IP}'$ denote the class obtained by changing the constant $1/3$ in the soundness condition for **IP** to 0. Prove that $\mathbf{IP}' = \mathbf{NP}$.

(c) Let $\mathbf{IP}'$ denote the class obtained by changing the constant $2/3$ in the completeness condition for **IP** to 1. Prove that $\mathbf{IP}' = \mathbf{IP}$.

Hint: use $\mathbf{IP} = \mathbf{PSPACE}$.

Problem 2: Easy Zero Knowledge. We defined Zero Knowledge for problems in the class **NP**, which includes **P** since $\mathbf{P} \subseteq \mathbf{NP}$. Give a trivial zero knowledge protocol for any problem in **P**. Your protocol will require no communication between parties. Prove that this protocol satisfies completeness, soundness, and perfect zero knowledge.

Problem 3: PCPs.

(a) Prove that any language L that has a PCP-verifier using r coins and q *adaptive* queries (where each query depends on responses to prior queries) also has a standard, i.e., nonadaptive, verifier using r coins and 2^q queries.

(b) Prove that $\mathbf{PCP}(0, \log n) = \mathbf{P}$.

(c) Prove that $\mathbf{PCP}(0, \mathsf{poly}(n)) = \mathbf{NP}$.

Problem 4: Approximating 3SAT. In this problem, we will give a probabilistic polynomial time algorithm that, given a 3CNF formula φ with exactly three distinct variables in each clause, outputs an assignment satisfying at least $7/8$ fraction of φ's clauses.

Note: although we will not prove it, it is possible to turn this algorithm into a deterministic algorithm with the same approximation.

(a) Show that a random assignment of values to the variables is expected to satisfy $7/8$ fraction of the clauses. You can prove this using *linearity of expectation*: for random variables X and Y, $E[X+Y] = E[X] + E[Y]$.

(b) Show that the probability of satisfying at least a $7/8 - 1/(2m)$ fraction of clauses, where m is the number of clauses, is at least $1/\mathsf{poly}(m)$. To do so, you can use the following concentration inequality, which can be derived from the Markov inequality:

If $X \in [0,1]$ and $E[X] = \mu$, then for any $c < 1$,

$$\Pr[X \leq c\mu] \leq \frac{1-\mu}{1-c\mu}$$

(c) Why does this prove that you have a probabilistic polynomial time algorithm for 3SAT?

Optional Feedback. Please answer the following questions to help design future problem sets. You are not required to answer these questions, and if you would prefer to answer anonymously, please use the anonymous feedback form. However, we do encourage you to provide feedback on how to improve the course experience.

(a) Roughly how long did you spend on this problem set?

(b) What was your favorite problem on this problem set?

(c) What was your least favorite problem on this problem set?

(d) Any other feedback for this problem set? Was it too easy/difficult?

(e) Any other feedback on the course so far?