CS 505 Spring 2025 — Homework 4

YOUR NAME HERE (FIRST AND LAST) (UID: YOUR UID HERE)

Due Date: April 15, 2025, no later than 2:00pm Central Time.

Collaboration Policy

Collaboration between students is encouraged. However, **all collaborations** need to be acknowledged (whether they are in this class or not in this class). You **MUST** list all collaborators for homework assignments. Moreover, collaborating **does not mean** you can copy-paste work from each other. Each submission needs to be in the students own words, otherwise it will be considered plagiarism.

Moreover, you are allowed to look to other resources for help with the homework. Please correctly cite such resources by using the `\cite` command, and including the correct citations in `local.bib`.

Finally, please acknowledge any other discussions that helped you complete this assignment. This can include "office hours," "Piazza," or other discussions where a direct collaboration did not happen.

Collaborator and Discussion Acknowledgements

Please list your collaborators below. Include their First and Last names, along with their UID if they are a UIC student. If you did not collaborate with others for this assignment, please copy-paste the following line into the first item of the `itemize` below.

I worked on this assignment individually and did not collaborate with others.

- Collaborator 1...

1 Boolean Circuits (25 Points)

1.1 Part 1 (5 Points)

Let $\mathrm{add}_n \colon \{0,1\}^{2n} \to \{0,1\}^{n+1}$ be the function $\mathrm{add}(x,y) = x + y$ for two n-bit integers x and y. Show that add_n is computable by an $O(n)$-sized circuit family. Here, the circuit has multiple outputs instead of $\{0,1\}$.

Proof of Problem 1 Part 1. Your answer here... ☐

1.2 Part 2 (10 Points)

A language $L \subseteq \{0,1\}^*$ is sparse if there exists a polynomial p such that for every $n \in \mathbb{N}$, $|L \cap \{0,1\}^n| \leq p(n)$. Show that every sparse language is in $\mathbf{P_{/poly}}$.

Proof of Problem 1 Part 2. Your answer here... ☐

1.3 Part 3 (10 Points)

Prove that a language L is decidable by a family of logspace uniform circuits if and only if $L \in \mathbf{P}$.

Proof of Problem 1 Part 3. Your answer here... ☐

2 Interactive Proofs (25 points)

2.1 Part 1 (5 Points)

Show that $\mathbf{AM}[2] = \mathbf{BP} \cdot \mathbf{NP}$.

Proof of Problem 2 Part 1. Your answer here... $\square$

2.2 Part 2 (10 Points)

The *graph isomorphism* problem defined as follows. Let $G_0 = (V_0, E_0), G_1 = (V_1, E_1)$ be two graphs, each on n vertices. We say that G_0 and G_1 are isomorphic if there exists a permutation $\pi\colon [n] \to [n]$ such that $\pi(G_0) = G_1$. That is, after relabeling the vertices of G_0 using the permutation π, we obtain the graph G_1. Another way to state this: if we permute the rows or columns of the adjacency matrix of G_0 according to π, we obtain the adjacency matrix for G_1.

Give an interactive proof for deciding if two graphs G_0, G_1 are isomorphic. In particular, answer the following questions.

1. What is the completeness error?
2. What is the soundness error?
3. How many rounds does the IP have?
4. If your soundness error bound is δ, how can you reduce it to δ^k?

Note: *I encourage you to try to solve this problem without consulting online sources first!*

Proof of Problem 2 Part 2. Your answer here... $\square$

2.3 Part 3 (10 Points)

Let $G = (V, E)$ be an undirected simple graph (i.e., no self loops) on n vertices. A *triangle* is a tuple of vertices $(i, j, k) \in V \times V \times V$ such that $(i, j), (j, k), (k, i) \in E$. Let

$$\mathrm{TRI} = \{(G, k) \colon G \text{ is a simple graph with } k \text{ triangles.}\}.$$

Show that $\mathrm{TRI} \in \mathbf{IP}$. Here, use the "standard" definition of $\mathbf{IP}$ presented in class (i.e., completeness and soundness error $\leq 1/3$).

Hint: consider the adjacency matrix view of G; let A be its adjacency matrix. View the matrix A as a Boolean function $f_A \colon \{0,1\}^{\log(n)} \times \{0,1\}^{\log(n)} \to \{0,1\}$, where $f_A(i,j) = 1$ if and only if $A[i,j] = 1$. How can you use f_A to count the number of triangles in a graph G with adjacency matrix A? Once you can do this, consider how to use the sum-check protocol in your interactive proof.

Proof of Problem 2 Part 3. Your answer here... $\square$