CS 535: Complexity Theory, Fall 2020

Homework 1

Due: 8:00PM, Friday, September 11, 2020.

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 (Collaboration Policy)**.** Read and sign the course Collaboration and Honesty Policy (https://cs-people.bu.edu/mbun/courses/535_F20/handouts/collaboration.pdf) and upload it to Gradescope. An electronic signature is fine.

Problem 2 (Random Pizza Party)**.** Uncle Mark's Pizza shop bakes pizzas with a huge number of possible toppings. You are hosting a pizza party for n very persnickety guests, numbered $1, \dots, n$. Every guest *likes* some set consisting of k toppings, and *dislikes* a disjoint set consisting of k toppings. A guest will only eat a pizza if it has all of the toppings they like and has none of the toppings they dislike. However, the guests don't eat much, so arbitrarily many guests can share any given pizza. Being a good host, you want to order a wide enough variety of pizzas so that every guest will have something to eat.

(a) Consider a random pizza that includes each possible topping independently with probability $1/2$. What is the probability that any particular guest i is willing to eat this pizza?

(b) Suppose you order m random pizzas independently according to the above process. What is the probability that guest i is willing to eat *none* of the pizzas you ordered? Your answer should only depend on k and m.

(c) Show that regardless of your guests' preferences, there exists a way to order $m = 2^{O(k)} \log n$ pizzas such that every guest will have something to eat.

Hint 1: Use the *probabilistic method*. Show that if you order $m = 2^{O(k)} \log n$ random pizzas, then the probability that there exists a guest who can eat none of the pizzas is strictly smaller than 1.

Hint 2: You may find the following inequality helpful: $(1 - 1/x)^x \leq \frac{1}{e}$ for all $x > 1$.

Problem 3 (Encodings)**.** An encoding is an unambiguous way to represent an object, e.g., a natural number, a graph, or a Turing machine, as a string in Σ^* for some alphabet Σ. Encodings allow us to present complex objects as inputs to Turing machines and other computational devices. Following Section 1.4 of Arora-Barak, one way to formalize a *nice* *encoding* of a set of objects D is as a function $f : \Sigma^* \to D$ such that, for every $y \in D$, there are infinitely many $x \in \Sigma^*$ such that $f(x) = y$. (Check that this indeed formalizes the discussion there!)

We will typically fix a reasonable, unspecified way of encoding objects as binary strings (strings over the binary alphabet $\Sigma = \{0, 1\}$) and refer to an object and its shortest encoding interchangeably. But it's sometimes worth keeping in mind how the choice of encoding affects the statements we can prove.

(a) Show that the set of Turing machines $\mathcal{M}$ can be nicely encoded in a unary alphabet $\Sigma = \{1\}$ via a function $f_1 : \{1\}^* \to \mathcal{M}$. You may use without proof the fact that Turing machines can be nicely encoded in binary as described in Section 1.4.

(b) A unary language L is a language over the alphabet $\{1\}$, i.e., $L \subseteq \{1\}^*$. Show that there is an undecidable unary language.