CS789: Introduction to Probabilistic Proof Systems Spring 2025

Assignment 1

February 5, 2025

Instructor: Chethan Kamath

Assignment Policy:

1. You have one week to submit the solutions (i.e, the deadline is 04/Feb, midnight).
2. Please use LaTeX to typeset up your solutions.
3. You are free to collaborate with others to solve the problems. But in the end you must *write up* the solutions on your own. Please list the persons you collaborated with on each problem.

Problem 1 ((3+3+2=8 points) Low-degree and multilinear extensions)**.**

1. In Lectures 1 and 2, we defined the Reed-Solomon encoding of a message $\bar{a} = (a_1, \ldots, a_n) \in \mathbb{F}_p^n$ as
$$\{f(0), \ldots, f(p-1)\},$$
where $f(x) := \sum_{i=1}^{n} a_i x^{i-1}$ and we assume $p \gg n$. Now let's consider encoding $\bar{a}$ as
$$\{f'(0), \ldots, f'(p-1)\},$$
where $f'(x)$ is the unique degree-$(n-1)$, univariate polynomial such that $f'(i) = a_{i+1}$ for all $i \in [0, n-1]$. f' is called the (univariate) *low-degree extension* of $\bar{a}$. Describe how to construct f' from $\bar{a}$. Prove that the low-degree extension – like Reed-Solomon encoding – is also *distance amplifying* in the sense that if two messages $\bar{a}$ and $\bar{a}'$ differ even in one position, their encodings differ in *many* positions (here, at least $p-n$ positions).

2. Now, we will extend the idea above to the *multilinear* setting. Consider a function $a : \{0,1\}^v \to \mathbb{F}_p$. A v-variate multilinear polynomial $\tilde{a} \in \mathbb{F}_p[X_1, \ldots, X_v]$ is a *multilinear extension* of a if $\tilde{a}(i) = a(i)$ for all $i \in \{0,1\}^v$ (where by $\tilde{a}(i)$, we mean $\tilde{a}(i_1, \ldots, i_v)$ for $i := i_1 \| \ldots \| i_v$). Describe how to construct $\tilde{a}$ from a. Prove that the multilinear extension is *also* distance amplifying.

3. In Lecture 5, we learned how to arithmetise a SAT formula $\varphi : \{0,1\}^n \to \{0,1\}$. Note that $\tilde{\varphi}$ constitutes an alternative way to arithmetise φ. What happens if you use the Sumcheck Protocol from Lecture 5 with $g(X_1, \ldots, X_n)$ set to $\tilde{\varphi}(X_1, \ldots, X_n)$?

Problem 2 ((2+2+2=6 points) Understanding the definition of interactive proof (IP))**.** Recall the definition of IP given in Definition 2, Lecture 3. In the following three sub-problems, we will tweak Definition 2 and then try to ascertain what happens to its expressivity.

1. Prove that the class remains unchanged (i.e., is **IP**) if the honest prover P is allowed to be *randomised*. (Hint: show that any randomised prover P can be converted into a deterministic prover P' by fixing the random coins of P appropriately.)

2. Does the class change if the malicious prover P^* is allowed to be randomised?

3. Show that the class of problems that have a *deterministic-verifier* IP is **NP**.

Problem 3 ((3 points) Program checking)**.** A "checker" for a computational task f (i.e., a decision or search problem) is a probabilistic polynomial time machine C that, given *any* program F that is a claimed program for f and any input x^*, has the following behaviour.

1. Completeness: If F is a correct program for f (i.e., $\forall x : \mathsf{F}(x) = f(x)$) then

$$\Pr\left[\mathsf{C}^{\mathsf{F}} \text{ accepts } (x^*, \mathsf{F}(x^*))\right] \geq 2/3\right],$$

where C^{F} denotes that C has oracle access to F (and thus can invoke F on inputs of its choice).

2. Soundness: If $\mathsf{F}(x^*) \neq f(x^*)$ then

$$\Pr\left[\mathsf{C}^{\mathsf{F}} \text{ accepts } (x^*, \mathsf{F}(x^*))\right] \leq 1/3.$$

Using ideas from IP for graph non-isomorphism (GNI) from Lecture 3, design and analyse a checker for GNI.

Problem 4 ((2+3+2=7 points) Sumcheck, in fewer rounds?)**.** In Lecture 5, we saw how Sumcheck Protocol, Π_{SC}, allows a prover to convince a verifier that

$$\sum_{a_1,\ldots,a_n \in \{0,1\}} g(a_1, \ldots, a_n) = K$$

using $2n$ rounds of interaction. In the following sub-problems assume, for simplicity, that n is a power of 2.

1. Design a protocol that reduces the number of rounds required to $2(n - \log(n))$. Analyse the soundness and completeness of your protocol.

2. Do you think it is possible to reduce the number of rounds required to $2n/\log(n)$? Either describe an $2n/\log(n)$-round protocol, or reason why this might not be possible.

3. Consider the following modification of Π_{SC}, which reduces the number of rounds to just two:

   (a) In round 1, V sends $(r_1, \ldots, r_n)$ to P, where the r_is are sampled as in Π_{SC}.

   (b) In round 2, P sends $(h_1(X_1), \ldots, h_n(X_n))$, where the $h_i(X_i)$s are computed as in Π_{SC}.

   Describe a cheating prover strategy that breaks soundness of this protocol.

Problem 5 ((3+3=6 points) Completing **IP** = **PSPACE**)**.** In Lecture 6, we saw that **PSPACE** $\subseteq$ **IP**. In the handwritten notes, a proof for **IP**[1] $\subseteq$ **PSPACE** is provided.

1. Understand that proof and then extend it to **IP**[2] $\subseteq$ **PSPACE**.

2. Use induction to then prove that **IP** $\subseteq$ **PSPACE**.