11. Assignment on

Computational Complexity Theory

SoSe 2016

Wolfgang Mulzer, Claudia Dieckmann

Due on 12. July 2016 in the tutorial session
This is the last problem set.

Problem 1 Goldwasser-Sipser Set Lowerbound *10 points*

(a) Let $n \in \mathbb{N}$. Define $\mathcal{H}_{n,n} := \{h_{a,b}\}_{a,b \in \mathrm{GF}(2^n)}$, where for $a, b \in \mathrm{GF}(2^n)$ we set $h_{a,b}(x) = ax + b$ for all $x \in \mathrm{GF}(2^n)$ (recall that $\mathrm{GF}(2^n)$ is the finite field with 2^n elements). Show that $\mathcal{H}_{n,n}$ is a pairwise independent family of hash functions.

(b) For $k \in \mathbb{N}$, define $\mathcal{H}_{n,k}$ as the family obtained from $\mathcal{H}_{n,n}$ by adding $k-n$ zeros to the input if $k > n$ or by dropping $n-k$ bits from the output if $k < n$. Show that $\mathcal{H}_{n,k}$ is a pairwise independent family of hash functions.

(c) Work out the details of the Goldwasser-Sipser protocol. In particular, use Chernoff-bounds to estimate the required number of samples.

Problem 2 Chernoff-Bounds *10 points*

(a) Read the notes on Chernoff-bounds on the website. Which proof technique do you like best? Why?

(b) Suppose we are given a coin with unknown probability $p > 0$ of coming up heads. Let $\varepsilon > 0$ and $\delta > 0$. Devise an algorithm that gives an *additive* ε approximation for p with probability at least $1 - \delta$ (i.e., we want an estimate $\tilde{p}$ such that $\Pr[|p - \tilde{p}| > \varepsilon] \leq \delta$). Repeat for the case of multiplicative error (i.e., $\Pr[|p - \tilde{p}| > \varepsilon p] \leq \delta$). What is the running time?

Problem 3 Set Lowerbound with Perfect Completeness *10 points*

(This is Problem 8.5 in Arora-Barak.) Show that there exists a perfectly complete AM[$O(1)$] protocol for proving a lower bound on set size.

Hint: First note that in the Goldwasser-Sipser protocol, we can have the prover choose the hash function. Consider the easier case of constructing a protocol to distinguish between the case $|S| \geq K$ and $|S| \leq K/c$, where $c \geq 2$ can even be a function of the input size. If c is large enough, we can allow the prover to use *several* hash functions $h_1, \ldots, k_\ell$, and it can be proven that if ℓ is large enough, we will have $\bigcup_i h_i(S) = \{0,1\}^k$ for large S. The gap can be increased by considering instead of S the set S^z, i.e., the z times Cartesian product of S.