Homework for Computational Complexity
Spring 2020, UPC Barcelona
Instructor: Albert Atserias

Created: 15/06/2020
Last modified: 15/06/2020

Homework 4 - Solutions

Exercise 1 The condition '$P_{a,b,c}(x_i) = y_i$ for $i = 1, 2, 3$' can be phrased as three linear equations on the variables a, b, c. Concretely: $ax_i^2 + bx_i + c = y_1$ for $i = 1, 2, 3$. Solving for a, b, c in this system of equations we get

$$a = ((y_1 - y_2)(x_1 - x_3) - (y_1 - y_3)(x_1 - x_2))((x_1 - x_2)(x_1 - x_3)(x_2 - x_3))^{-1} \tag{1}$$
$$b = (y_1 - y_2)(x_1 - x_2)^{-1} - a(x_1 + x_2) \tag{2}$$
$$c = y_1 - ax_1^2 - bx_1 \tag{3}$$

where the inverses are mod p and they exist because p is prime and x_1, x_2, x_3 are distinct so the differences $x_i - x_j$ are non-zero mod p when $i \neq j$. This means that (a, b, c) is completely determined from the event '$P_{a,b,c}(x_i) = y_i$ for $i = 1, 2, 3$', which means that the probability of the event is $1/p^3$ since there are p^3 possible triples $(a, b, c) \in \mathbb{Z}_p$.

Exercise 2 In case $|S| > 2^k$, the pigeonhole principle implies that for any $h \in U_{m,k}$ there exists at least two distinct x and y in S with $h(x) = h(y)$. So $\Pr_h[I(S, h) = 1] = 1$. In case $|S| \leq 2^{k/2}$, we have

$$\Pr_h[I(S,h) = 1] = \Pr_h[\exists x, y \in S \; \exists z \in \{0,1\}^k (x \neq y \text{ and } h(x) = z \text{ and } h(y) = z)] \tag{4}$$
$$\leq \sum_{\substack{x,y \in S: \\ x \neq y}} \sum_{z \in \{0,1\}^k} \Pr_h[h(x) = z \text{ and } h(y) = z] \tag{5}$$
$$= \binom{|S|}{2} 2^k 2^{-2k} \tag{6}$$
$$= (2^{k/2}(2^{k/2} - 1)/2)2^{-k} \tag{7}$$
$$\leq 1/2. \tag{8}$$

Exercise 3 Let $f(x)$ be a #P-function, so $f(x) = |W(x)|$ where $W(x) \subseteq \{0,1\}^m$ with $m := p(|x|)$ for some polynomial p and the predicate '$y \in W(x)$' can be tested in time polynomial in n. Let $U_{m,k}$ be a 2-universal family with $2^{\mathrm{poly}(m,k)}$ many functions that can be computed in polynomial time. Given x of length n, we search for the largest $k = 1, 2, \ldots, 2m$ for which there exists $h \in U_{m,k}$ that avoids collisions on $W(x)$ (we will show below that such a k always exists, so a largest one always exists), and output $t := 2^{k/2}$. Whether there is an $h \in U_{m,k}$ that avoids collisions on $W(x)$ can be expressed as

$$\exists h \in U_{m,k} \; \forall z_1, z_2 \in \{0,1\}^m \; (z_1 \in W(x) \land z_2 \in W(x) \land z_1 \neq z_2 \rightarrow h(z_1) \neq h(z_2)).$$

This is can be tested with a Σ_2^{P}-oracle: each function in $U_{m,k}$ is specified by $\mathrm{poly}(m,k) \leq \mathrm{poly}(n)$ many bits, the z_1 and z_2 take $k \leq m \leq \mathrm{poly}(n)$ many bits, and the condition in the matrix of the formula can be checked in polynomial time. Therefore, the algorithm can be implemented with at most $p(n)$ many queries to an oracle in Σ_2^{P} (by binary search we can even reduce this to $O(\log(n))$ many queries, but this is not very important for us). By part 2 of the previous exercise we known that the output $t = 2^{k/2}$ satisfies $|W(x)| > 2^{(k+1)/2} = t\sqrt{2}$, since whenever $|W(x)| \leq 2^{(k+1)/2}$ there is at least one $h \in U_{m,k+1}$ that avoids collisions (in fact half such h do). Since $|W(x)| \leq 2^m$, in particular this shows that a largest $k \in \{1, 2, \ldots, 2m\}$ as wanted by the algorithm always exists. By part 1 of the previous exercise we know that the output $t = 2^{k/2}$ satisfies $|W(x)| \leq 2^k = t^2$, since whenever $|W(x)| > 2^k$ there is no way any $h \in U_{m,k}$ can avoid collisions. Thus, the output satisfies $t \leq t\sqrt{2} < f(x) \leq 2^k = t^2$.