Theoretische Informatik 2 Exercises

Exercise 32

Given: $f : \{0,1\}^* \to \{0,1\}^*$ one-way permutation

Task: Show that f^k is one-way $\forall k \in \mathbb{N}$

Proof. by induction on k.

- $k = 1$
- $k > 1$

  $k = 2$

  By hypothesis, we have that for every probabilistic polynomial time algorithm A, the following holds $\forall n \in \mathbb{N}$:

  $$\mathbb{P}(f(A(f(x))) = f(x)) < \varepsilon(n),\ x \in \{0,1\}^n$$

  Consider an arbitrary probabilistic polynomial algorithm B. Observe that

  $$\mathbb{P}(f^2(B(f^2(x)) = f^2(x)) = P(f(f(B(f(f(x))))) = f(f(x)))$$

  A permutation is bijective, so there exists an inverse function $f^{-1} \to$ apply f^{-1} on both sides and yield

  $$\mathbb{P}(f(B(f(f(x)))) = f(x)) = \mathbb{P}[f(\underbrace{B \circ f(x))}_{\text{a prob. polyn. alg.}} = f(x)] < \varepsilon$$

  $\Rightarrow$ defining $A := B \circ f$ we can show the assumption

$\square$

Exercise 34

a. Prove that $\mathrm{PCP}(0, \log n) = \mathrm{P}$

- "$\mathrm{P} \subseteq \mathrm{PCP}(0, \log n)$"

  Let $L \in \mathrm{P}$. A verifier V of $\mathrm{PCP}(0,0) \subseteq \mathrm{PCP}(0, \log n)$ has polynomial running time and can decide L.
- "$\mathrm{P} \supseteq \mathrm{PCP}(0, \log n)$"

  Algorithm:
  For each proof $(O(2^{\log n}) = O(n))$ If V accepts, accept Else Reject
  Total running time: $O(n) \cdot \mathrm{poly}(n) = \mathrm{poly}(n)$

b. Prove that $\mathrm{PCP}(0, \mathrm{poly}(n)) = \mathrm{P}$

- There is a verifier V polynomial, deterministic
- V decides L
- $P(...) < \frac{1}{2}$ means 0 (no random bits)

Exercise 35

Show that $\mathrm{PCP}(\log n, 1) \subseteq \mathrm{NP}$.

Proof. Let $L \in \mathrm{PCP}(logn, 1)$. We build a non-deterministic TM M which works as follows:

1. M generates non-deterministically all the proofs of length at most $\cdot 2^{O(\log n)}$. This can be done in $O(\log n)$ steps.
2. M generates non-deterministically all the $2^{O(\log n)}$ possible sequences of coin tosses
3. M emulates the verifier on all these toss sequences ($M \in \mathrm{PCP}$)
4. M accepts $\Leftrightarrow$ the verifier accepts on all these sequences

$\rightarrow M$ runs in $2^{O(\log n)} = \bigcup_{c \geq 0} n^c \Rightarrow L \in \mathrm{NP}$

$\square$

Exercise 37

Task: Provide a $\mathrm{PCP}(\mathrm{poly}(n, 1)$ verifier for the complement of the graph isomorphism problem.
$\overline{\mathrm{GI}}$ is the complement of GI, i.e. the language consisting of non-isomorphic graphs.

Input: graphs G_0, G_1 which both have $n$vertices and m edges.
The verifier expects the proof Π to contain a bit $\Pi(H)$ ($\in \{0,1\}$) for each labeled graph with n nodes such that $\Pi(H) \in \{0,1\}$ corresponding to whether $H \cong G_0$ or $H \cong G_1$

$\rightarrow$ in other words, Π can be seen as an exponentially long array of bits indexed by all possible graphs on n vertices.
Verifier picks random bit $b \in \{0,1\}$ and a random permutation $\rho \in S_n$

Apply ρ to vertices of G_b.
$\rightarrow$ Leads to graph $H \cong G_b$
Verifier queries the proof bit $\Pi(H)$ and accepts if this bit equals b

Case 1: $G_0 \ncong G_1$
In this case Π can be set up such that the verifier accepts with probability 1

Case 2: $G_0 \cong G_1$
The probability that any proof makes the verifier accept is at most $\frac{1}{2}$

Exercise 39

$$(x_1 \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee \overline{x_3} \vee x_4) \wedge (x_2 \vee x_3 \vee \overline{x_4})$$

a.

$$q = (1-x_1) \cdot x_2 \cdot (1-x_3) + x_1 \cdot x_3 \cdot (1-x_4) + (1-x_2) \cdot (1-x_2) \cdot x_4$$

$$= x_2 - x_1 \cdot x_2 - x_2 \cdot x_3 + x_1 \cdot x_2 \cdot x_3 + x_1 \cdot x_3 - x_1 \cdot x_3 \cdot x_4 + x_4 - x_2 \cdot x_4 - x_3 \cdot x_4 + x_2 \cdot x_3 \cdot x_4$$

$$= x_2 + x_4 - x_1 \cdot x_2 + x_1 x_3 - x_2 x_3 - x_2 x_4 + x_1 x_2 x_3 - x_1 x_3 x_4 + x_2 x_3 x_4$$

b.

$$I_q^1 = \{2,4\}$$

$$I_q^2 = \{(1,2),(1,3),(2,3),(2,4),(3,4)\}$$

$$I_q^3 = \{(1,2,3),(1,3,4),(2,3,4)\}$$

c.

$$a = (1,0,1,1)$$

$$\gamma + L_1^q(a_1^1) + L_2^q(c_1^2) + L_2^(c_q^3)$$

$$C_{q_i}^1 = \begin{cases} 1 & i \in I_1^1 \\ 0 & i \notin I_1^1 \end{cases}$$

$$\gamma_q = 0$$

$$L_1^a(c_q^1) = a_2 + a_4 = 1$$

$$L_2^a(c_q^1) = a_1a_2 + a_1a_3 + a_2a_3 + a_2a_4 + a_3a_4 = 0(2)$$

$$L_3^a(c_q^1) = a1a_2a_3 + a_1a_3a_4 + a_2a_3a_4 = 1$$

$$\sum = 0 + 1 + 0 + 1 = 0.(2)$$

Insert into the original formula

$0 + 0 + 0 = 0$