PS Complexity of Polynomial-Time Problems

https://www.cosy.sbg.ac.at/~sk/courses/polycomp/

Exercise sheet 2 Due: Sunday, November 19, 2017

Total points : 40

Prove all your claims!

Exercise 1 (*10 points)*

Prove the following: If there is a decision algorithm to check whether for a given pair of sets $A, B \in \{0,1\}^d$ there exists a pair of vectors $a \in A, b \in B$ such that $a \perp b$ with running time $O(n^{2-\epsilon}\,\mathrm{poly}(d))$ (for some constant $\epsilon > 0$), then there also is an algorithm with running time $O(n^{2-\delta}\,\mathrm{poly}(d))$ (for some constant $\delta > 0$) that, given pair of sets $A, B \in \{0,1\}^d$, ouputs a pair $a \in A, b \in B$ such that $a \perp b$ if such a pair exists.

Exercise 2 (*10 points)*

Consider the **Longest Palindromic Subsequence** problem: Given a sequence S of length n, find the longest common subsequence which is a palindrome (i.e., a sequence of characters which reads the same forward and backward).

Prove that, assuming **OVH**, there is no algorithm for **Longest Palindromic Subsequence** with running time $O(n^{2-\epsilon})$ (for any constant $\epsilon > 0$).

Exercise 3 (*10 points)*

Consider the **Hitting Set** problem: Given two lists of n subsets over a universe U of size $d = |U|$, determine if there is a set in the first list that intersects every set in the second list, i.e. a "hitting set".

The **HSH** (Hitting Set Hypothesis) states that **Hitting Set** admits no algorithm with running time $O(n^{2-\epsilon} \cdot \mathrm{poly}(d))$ (for any constant $\epsilon > 0$). Prove that **HSH** implies **OVH**.

***Hint**: Be careful about the direction this reduction has to go. In the lecture, we gave a reduction from **All-Pairs Negative Triangle** to **Negative Triangle**. The same type of reduction will work here.*

Exercise 4 (*10 points)*

Recall the following formal definition of *subcubic reductions*: Let $\mathbf{P}$ and $\mathbf{Q}$ be computational problems with a common size measure n on the inputs. We say that there is a subcubic reduction from $\mathbf{P}$ to $\mathbf{Q}$ if there is an algorithm $\mathscr{A}$ with oracle access to $\mathbf{Q}$ satisfying three properties:

- For every instance x of $\mathbf{P}$, $\mathscr{A}(x)$ solves the problem $\mathbf{P}$ on x.
- Excluding oracle calls, $\mathscr{A}$ runs in time $O(n^{3-\gamma})$ (for some constant $\gamma > 0$) on instances of size n.
- For every constant $\varepsilon > 0$ there is a constant $\delta > 0$ such that for every instance x of $\mathbf{P}$ of size n we have $\sum_i n_i^{3-\varepsilon} \leq n^{3-\delta}$, where n_i is the size of the ith oracle call to $\mathbf{Q}$ in $\mathscr{A}(x)$.

We use the notation $\mathbf{P} \leq \mathbf{Q}$ to denote the existence of a subcubic reduction from $\mathbf{P}$ to $\mathbf{Q}$. Prove that subcubic reductions are transitive. In other words, prove that if $\mathbf{P} \leq \mathbf{Q}$ and $\mathbf{Q} \leq \mathbf{R}$ then $\mathbf{P} \leq \mathbf{R}$.