max planck institut
informatik

SAARLAND
UNIVERSITY
COMPUTER SCIENCE

Karl Bringmann and Sebastian Krinninger **Summer 2016**

Exercises for Complexity Theory of Polynomial-Time Problems

`https://www.mpi-inf.mpg.de/departments/algorithms-complexity/teaching/summer16/poly-complexity/`

Exercise sheet 2 Due: **Monday, May 30, 2016**

Total points : 40

Either email an electronic version of your assignment submission to gjindal@mpi-inf.mpg.de or give it to Gorav in his office at Room 425, Building E1.3. If Gorav is not in his office then you can just slide your submission under the door of his office.

You are allowed to collaborate on the exercise sheets, but you have to write down a solution on your own, using your own words. Please indicate the names of your collaborators for each exercise you solve. Further, cite all external sources that you use (books, websites, research papers, etc.).

You need to collect at least 50% of all points on exercise sheets.

Exercise 1 (*6 points*) The Hitting Set Problem is defined as follows: Given two lists of n subsets over a universe U of size d, determine if there is a set in the first list that intersects every set in the second list, i.e. a “hitting set”.

Define a Boolean circuit C that outputs 1 if and only if there is a hitting set, using suitable encodings of the inputs.

Exercise 2 (*13 points*)

a) (5 points) *Do not be scared by the long statement of this exercise, answer to this exercise would be shorter than the statement.*
Recall the following definition from the lecture of approximation/representation of boolean circuits by polynomials.

Definition (Polynomial representation of Circuits). *Let C be a Boolean circuit with k input gates and let $\mathcal{D}$ be a finite distribution of polynomials on k variables over a ring R containing 0 and 1. The distribution $\mathcal{D}$ is a probabilistic polynomial over R representing C with error δ if for all $(x_1, x_2, \ldots, x_k) \in \{0,1\}^k$:*

$$\Pr_{p\sim\mathcal{D}}[p(x_1, x_2, \ldots, x_k) = C(x_1, x_2, \ldots, x_k)] \geq 1 - \delta.$$

In the lecture we saw how polynomials $P(x_1, x_2, \ldots, x_k)$ of degree t can approximate/represent (with success probability $1-\frac{1}{2^t}$) the logical OR function $OR(x_1, x_2, \ldots, x_k) = x_1 \vee x_2 \vee \ldots \vee x_k$.

Let us define a new polynomial $Q(x_1, x_2, \dots, x_k)$ which equals to 1, i.e, $Q(x_1, x_2, \dots, x_k) = 1$. Now note that $Q(x_1, x_2, \dots, x_k)$ computes the OR function on all inputs except the input $(0, 0, \dots, 0)$. So it fails to compute the OR function only on a $\frac{1}{2^k}$ fraction of all possible inputs and this polynomial has degree 0. In other words, if we take a random input then this polynomial $Q(x_1, x_2, \dots, x_k)$ computes the OR function with success probability $1 - \frac{1}{2^k}$. Why does this polynomial not represent/approximate the OR function as in the above definition? And why did we choose the above definition for polynomial representation/approximation of circuits?

b) (8 points) Recall that the above polynomial $P(x_1, x_2, \dots, x_k)$ has $(k+1)^t$ many monomials. In the lecture, we also saw a polynomial $R(x_1, x_2, \dots, x_k) = 1 \oplus \prod_{i=1}^{k}(1 \oplus x_i)$, which computes the OR function exactly. $R(x_1, x_2, \dots, x_k)$ has $2^k - 1$ monomials. Show that any polynomial computing the OR function *exactly* must have $2^k - 1$ monomials, i.e, $R(x_1, x_2, \dots, x_k)$ is the optimal polynomial to compute the OR function exactly. (This shows that we really do need approximation/representation if we want to compute boolean functions by polynomials having a low number of monomials.)

Hint**: First show that without loss of generality we can assume that any minimal polynomial computing the OR function has to be multilinear (a polynomial is multilinear if the degree of every variable in the polynomial is at most 1). Now show a bijection between the* 2^{2^k} ***boolean functions from $\{0,1\}^k$ ***to*** $\{0,1\}$ *and the* 2^{2^k} ***multilinear polynomials in*** $\mathbb{F}_2[x_1, x_2, \dots, x_k]$*. For this you need to show that for each boolean function from* $\{0,1\}^k$ *to* $\{0,1\}$*, there exists a unique multilinear polynomial in* $\mathbb{F}_2[x_1, x_2, \dots, x_k]$ *which computes this boolean function exactly. Finally, count the monomials in* $R(x_1, \dots, x_k)$*.*

Exercise 3 (*12 points*)

a) (7 points) Consider the following problem:
Max Inner product: Given two sets $A, B \subseteq \mathbb{Z}^d$ such that $|A| = |B| = n$. Find the value of the max inner product between any vector a from A and b from B, i.e, compute the quantity $\max_{a \in A, b \in B} \langle a, b \rangle$.
Prove that if **Max Inner product** can be solved in time $O(n^{2-\epsilon} \cdot \text{poly}(d))$ then **OVH** fails.

b) (5 points) Prove that if **OVH** fails then we can also find (if it exists) in time $O(n^{2-\epsilon} \cdot \text{poly}(d))$ a pair of vectors $a \in A, b \in B$ such that $a \perp b$, given two sets $A, B \subseteq \{0,1\}^d$ such that $|A| = |B| = n$.

Exercise 4 (*9 points*) Consider the following problem:
Longest Palindrome Subsequence: 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 backward or forward).

Prove that if **Longest Palindrome Subsequence** can be solved in time $O(n^{2-\epsilon})$ then **OVH** fails.