

Problem 1

Recall that an *arithmetic formula* $F(x_1, \dots, x_n)$ over the integers is an expression like

$$(x_1 + 4 \times x_3 \times x_4) \times ((x_2 - x_3) \times (x_2 + x_3)) - 3 \times x_2.$$

Just like boolean circuit lower bounds, arithmetic formula lower bounds are considered very difficult to prove.

In class we saw that the polynomial identity testing problem (for arithmetic formulas) is in the class coRP. Here you will show that if there is a deterministic algorithm for this problem, then there is a problem $f \in \text{NEXP}$ such that its multilinear extension $ML(f)$ over the integers cannot be computed by any polynomial-size family of arithmetic formulas.

Assume that this is false, so both of these assumptions hold:

- 1 There is a deterministic algorithm for polynomial identity testing
- 2 For every $f \in \text{NEXP}$, $ML(f)$ can be computed by a polynomial-size family of arithmetic formulas.

You will derive a contradiction.

- (a) Using assumption 2, show that the permanent polynomials

$$\text{per}_n(x_{i,j})_{1 \leq i,j \leq n} = \text{per}_n(x_{11}, \dots, x_{nn}) = \sum_{\pi} \prod_{i=1}^n x_{i\pi(i)}$$

have polynomial size arithmetic formulas.

- (b) Argue that the family of permanent polynomials is the unique family of polynomials p_1, p_2, \dots that satisfies the system of equations

$$p_n(x_{ij})_{1 \leq i,j \leq n} = \sum_{k=1}^n x_{1k} \cdot p_{n-1}(x_{ij})_{1 \leq i,j \leq n, i \neq 1, j \neq k}$$

$$p_{n-1}(x_{ij})_{1 \leq i,j \leq n-1} = p_n(x_{ij}^*)_{1 \leq i,j \leq n}$$

where

$$x_{ij}^* = \begin{cases} x_{ij}, & \text{if } 1 \leq i, j \leq n-1, \\ 1, & \text{if } i = j = n, \\ 0, & \text{otherwise.} \end{cases}$$

and $p_1(x_{11}) = x_{11}$.

- (c) Using assumption 1 and part (b), show that $P^{\#\text{SAT}} = \text{NP}$.
- (d) Using assumption 2 show that $\text{NEXP} = \text{MA}$.
- (e) Recall Toda's theorem which says that $\text{PH} \subseteq P^{\#\text{SAT}}$. Show that this is inconsistent with parts (c) and (d).

Problem 2

In this problem you will show that if we don't put reasonable restrictions on the class of ensembles, average-case complexity is no easier than worst-case complexity.

- (a) Show that for every $L \notin \text{P}$ there exists an ensemble μ_L such that (L, μ_L) does not have polynomial-time heuristic algorithms. (**Hint:** μ_L should give a lot of weight to the "hard" instances of L .)
- (b) Show that there exists an ensemble μ such that for every $L \in \text{NP}$, (L, μ) has polynomial-time heuristic algorithms if and only if $L \in \text{P}$. (**Hint:** Use the various μ_L from part (a) to construct μ .)

Problem 3

Show that (L, μ) has an average polynomial-time algorithm if and only if there is an algorithm A with the following properties:

- A takes two inputs x and ϵ and runs in time $\text{poly}(|x|, 1/\epsilon)$.
- For every input x and every ϵ , $A(x, \epsilon)$ outputs either $L(x)$ ("yes" if $x \in L$, "no" if $x \notin L$) or the special symbol "fail".
- For every n and ϵ ,

$$\Pr[A(x, \epsilon) = \text{"fail"}] \leq \epsilon.$$

Using this alternative definition of average polynomial-time algorithms, conclude that if (L, μ) reduces to (L', μ') and (L', μ') has an average polynomial-time algorithm, so does (L, μ) .