

PROBLEM 1 (20 Points) - Simulations Between Classes

1. Prove that any algebraic formula Φ of size s can be simulated by an ABP of size $O(s)$.
2. Prove that an ABP of size s can be simulated by a circuit of size $\text{poly}(s)$.

PROBLEM 2 (20 Points) - Another Proof that Permanent in VNP

In this question we will study the complexity of certain polynomials with $\{0, 1\}$ coefficients.

1. Given a 3-CNF formula Φ over the boolean variables w_1, \dots, w_n , define the polynomial $f_\Phi(x_1, \dots, x_n)$ as the polynomial whose coefficient of the monomial $x_1^{w_1} \cdots x_n^{w_n}$ equals $\Phi(w_1, \dots, w_n)$.
Prove that $f_\Phi \in \text{VNP}$.
2. Generalize the statement of part (1) to hold if we have Φ being any boolean formula of polynomial size.
3. Use part (2) to conclude that the permanent polynomial $Per \in \text{VNP}$.

PROBLEM 3 (20 Points) - Some Families in VP

1. Consider the elementary symmetric polynomial of degree d in n variables.

$$ES_d(x_1, \dots, x_n) = \sum_{S \subset [n], |S|=d} \prod_{i \in S} x_i$$

Prove that for any pair (n, d) where $n \geq d$ the elementary symmetric polynomial can be computed by a circuit of size $\text{poly}(n, d)$ and depth 3. That is, the elementary symmetric polynomials can also be computed really fast in the parallel model.

Hint: Consider the polynomial $p(x_1, \dots, x_n, t) = \prod_{i=1}^n (t + x_i)$

2. The Schur polynomials are an important family of polynomials in algebraic combinatorics and representation theory. These polynomials generalize the elementary symmetric polynomials as well as the complete homogeneous symmetric polynomials.

They are defined as follows: given a partition $\lambda := (\lambda_1, \lambda_2, \dots, \lambda_n)$ where $\lambda_1 \geq \dots \geq \lambda_n \geq 0$, we have

$$s_\lambda(x_1, \dots, x_n) := \sum_{T \in ST(\lambda)} \prod_{i=1}^n x_i^{t_i}$$

where $ST(\lambda)$ is the set of all semi-standard Young Tableaux of shape λ and t_i counts the number of occurrences of the element i in the tableau T .

Prove that for any partition λ of d into n parts, s_λ can be computed by a circuit of size $\text{poly}(n, d)$.

Hint: you do not need to know anything about tableaux to solve this problem! You may want to look up https://en.wikipedia.org/wiki/Schur_polynomial for an alternative way to write these polynomials and use division elimination.

PROBLEM 4 (20 Points) - Homogenization via Interpolation

In lecture we saw how to efficiently compute the homogeneous component of a circuit of any degree, by transforming the circuit. Here we will now see how to efficiently compute the homogeneous component of a circuit via interpolation.

Let $\Phi(x_1, \dots, x_n)$ be a circuit of size s computing a polynomial $f(x_1, \dots, x_n)$ of degree d . Let

$$f(x_1, \dots, x_n) = \sum_{i=0}^d f_i(x_1, \dots, x_n)$$

be the decomposition of f into its homogeneous components.

Let t be a new formal variable and consider the polynomial $f(tx_1, \dots, tx_n)$ computed by $\Phi(tx_1, \dots, tx_n)$.

Use interpolation to extract the degree k homogeneous component of f from the polynomial computed by $\Phi(tx_1, \dots, tx_n)$.

PROBLEM 5 (20 Points) - Matrix Multiplication and Circuit Complexity

Let X, Y be two $n \times n$ matrices. We say that a circuit Φ computes the product of X and Y if Φ contains n^2 gates g_{ij} such that g_{ij} computes the dot product of the i -th row of X and the j -th column of Y . As we have discussed in class, such circuits capture all algebraic ways to compute the product of two matrices.

Let ω be the exponent of matrix multiplication, where ω is defined as follows:

$$\omega := \inf_{\alpha \in \mathbb{R}} \{ \exists O(n^\alpha) \text{ size algebraic circuit computing } XY \}.$$

Let $X = (x_{ij}), Y = (y_{ij})$ and $Z = (z_{ij})$ be three symbolic $n \times n$ matrices.

Let

$$P(X, Y, Z) := \sum_{i,j=1}^n z_{ij} \sum_{k=1}^n x_{ik} y_{kj}.$$

Note that the coefficient of z_{ij} in P is the dot product of the i -th row of X and the j -th column of Y . That is, it computes the i, j -th entry of the product of X and Y .

1. Let Φ be a circuit of size s computing P . Show that $\omega \leq \log_n(s)$.
2. Let Ψ be a circuit of size $s := n^\alpha$ computing the product of X and Y . Show that Ψ can be transformed into a circuit of size $O(s)$ computing P .

Thus, we have reduced the problem of computing the exponent of matrix multiplication to the problem of computing the size of a circuit computing a specific polynomial.