**CS760 Topics in Computational Complexity** **Jul-Nov 2024**

# Assignment 2

*Total Marks: 50* *Deadline: Oct 22, Tuesday, 5 pm*

Note: Please write your answers precisely and succinctly. You are not supposed to discuss the problems with anyone else. If you need hints/clarifications, ask on Piazza or in the class.

1. (10 marks) For a CNF formula let its size be sum of the sizes of its clauses, where the size of a clause is the number of literals in it. For example $(x_1 \vee \neg x_2) \wedge (x_1 \vee x_3)$ has size 4. Suppose there is a polynomial time computable function $f : \{0,1\}^* \to \{0,1\}^*$ with the following property: For every input size $s > 0$, there is a set $T_s \subseteq \{0,1\}^*$ with $|T_s| \leq s^5$ such that for any formula $\phi$ of size at most $s$,

$$\phi \text{ is satisfiable if and only if } f(\phi) \notin T_s.$$

   Then prove that P=NP. Note that we do not know the set $T_s$.

   Hint: given a CNF formula, do a depth-first search for finding a satisfying assignment. Use function $f$ to keep the search space polynomially bounded.

2. (10 marks) In the class, we proved that determinant is a signed sum over clow sequences. Describe a parallel algorithm for computing this sum, which takes $O(\log^c n)$ time for some constant $c$. You can assume that polynomially many tasks can be done in parallel, if they are not dependent on each other.

3. Recall the combinatorial algorithm we discussed for determinant. Any matrix can be viewed as a weighted complete directed graph (with self-loops on every vertex), where edge weights are corresponding matrix entries. A cycle cover in a directed graph is a sequence of pairwise vertex-disjoint cycles that covers all vertices. A cycle, by definition, does not allow repetition of vertices (except starting and ending vertices being same). A self-loop is a cycle of length 1.

   Let us now consider a variant of cycle cover. A partial cycle cover is a sequence of pairwise vertex-disjoint cycles (that need not cover all vertices). A loop is defined to be a closed walk (starting and ending at the same vertex, vertex and edge repetitions allowed). We **do not** require that the starting vertex of a loop is the minimum among all vertices in the loop. That means $(u_1, u_2, u_3, u_1)$ and $(u_2, u_3, u_1, u_2)$ are considered two different loops. For a loop $L$ with edge sequence $(e_1, e_2, \ldots, e_q)$, its length is $q$ and its weight is $w(L) = \prod_{i=1}^{q} w(e_i)$. For a partial cycle cover, its length is sum of the lengths of the cycles in it and weight is the product of weights of cycles. A loop-cycle cover is a partial cycle cover together with one loop (the loop may or may not share vertices/edges with the partial cycle cover).

   Let us define

$$pcc_j = \sum_{\substack{\mathcal{C} \text{ is a} \\ \text{partial cycle cover} \\ \text{of length } j}} (-1)^{\text{number of cycles in } \mathcal{C}} \times w(\mathcal{C}).$$

$$\ell_j = \sum_{\substack{L \text{ is a} \\ \text{loop} \\ \text{of length } j}} w(L).$$

   (a) (4 marks) Show that $\ell_j$ can be computed in polynomial time for every $1 \leq j \leq n$.

   (b) (4 marks) Show that $pcc_j$ can be recursively computed using $pcc_1, pcc_2, \ldots, pcc_{j-1}$ and $\ell_1, \ell_2, \ldots, \ell_n$.

   (c) (2 marks) Clearly, $det = (-1)^n \cdot pcc_n$. What is the time complexity of this algorithm computing determinant.

4. (a) (2 marks) Argue that any CNF or DNF computing the parity of $n$ variables must have width $n$.

   In the class, we had seen a proof idea of the following statement.

   Let $f$ be a DNF of width at most $w$ over $n$ variables. Let $\alpha$ be a random restriction with $s = \sigma n$ stars, where $\sigma \leq 1/5$. Then for each $d \geq 0$ (and $s$),

$$\Pr[\text{DTdepth}(f|_\alpha) > d] \leq (10\sigma w)^d.$$

   (b) (5 marks) Assume the above statement. Suppose there is a depth-3 formula (AND-OR-AND, fan-out=1) computing the parity function on $n$ variables, whose bottom layer (just above input gates) has width at most $\sqrt{n}$. Then prove that the size of the formula is at least $2^{\sqrt{n}/40}$. The constant 40 is not important, any constant is fine.

   (c) (3 marks) Suppose there is a depth-3 formula (AND-OR-AND, fan-out=1) computing the parity function on $n$ variables. Then prove that the size of the formula is at least $2^{\sqrt{n}/80}$. The constant 80 is not important, any constant is fine.

   Hint: use random restrictions to convert the formula to small width. If convenient, you can set each variable independently as star/0/1.

5. Suppose there is a language $L$ for which there is a log-space probabilistic Turing machine $M$ (work tape has $O(\log n)$ space) such that for each $x \in \{0,1\}^*$,

$$\Pr[M(x) = L(x)] \geq 2/3.$$

   Find a polynomial time algorithm for the language $L$.