15-855: An Introduction to Computational Complexity Theory V. Guruswami & M. Wootters
Problem Set #5 Fall 2015
Due by 5 pm, Wednesday, **November 25, 2015** (in person or by email to yuzhao1@cs.cmu.edu).

Instructions: Same as for problem set 1.

Solve any 5 out of the 6 problems. Note that just because a problem statement is long, doesn't mean it is hard or has a long solution; in fact often it is the contrary. *If you turn in solutions to more than 5 problems, we will take your top scoring 5 problems.*

1. We begin with two short exercises about the Parity function:

   (a) **"Isn't Parity $\notin \mathsf{AC}^0$"?** Show that there is an AC^0 circuit with n output bits and poly(n) input bits such that when the input bits are chosen uniformly at random, the output bits are distributed uniformly on the set $\{x \in \{0,1\}^n : x_1 \oplus x_2 \oplus \cdots \oplus x_n = 1\}$.

   (b) **"Parity $\leq_{\mathsf{AC}^0}$ Majority."** Construct an AC^0-with-unbounded-fanin-Majority-gates circuit which computes Parity. (Hint: you can even do it in depth 2, with only negation and Majority gates, plus wires hard-coded to 0's and 1's.) Deduce a size lower bound for depth d unbounded-fanin AND-OR circuits computing the Majority function on n bits.

2. **"Majority $\notin \mathsf{AC}^0(\oplus)$."** While Majority gates are helpful for computing Parity by the above exercise, it turns out that Parity gates are not helpful to compute majority. Let $\mathsf{AC}^0(\oplus)_d$ be the class of circuits of depth-d with unbounded fan-in $\vee$, $\wedge$, and $\oplus$ gates (we assume negated forms of all variables are available at the leaves). Your goal is to work out a polynomial-based approximation approach similar to lecture to prove that computing Majority of n bits requires $\mathsf{AC}^0(\oplus)_d$ circuits of size $\exp(\Omega(n^{\frac{1}{2d}}))$. Instead of polynomials over reals as in class, in this problem we will work with polynomials over $\mathbb{F}_2$, the field with two elements $0, 1$ (so the addition and multiplication operations will be modulo 2).

   (a) Let $f : \{0,1\}^n \to \{0,1\}$ be computed by an $\mathsf{AC}^0(\oplus)_d$ circuit with S gates. Fix $\epsilon \in (0, 1/4)$. Prove that there is a multilinear polynomial $p \in \mathbb{F}_2[X_1, X_2, \ldots, X_n]$ of total degree at most $O((\log(S/\epsilon))^d)$ such that $f(a) = p(a)$ for at least $1 - \epsilon$ fraction of $a \in \{0,1\}^n$.

   (Hint: Follow the gate-by-gate approach from lecture. Working over $\mathbb{F}_2$ in fact makes things easier.)

   (b) Define the subset $S_0 \subset \{0,1\}^n$ (resp. $S_1 \subset \{0,1\}^n$) to be those inputs for which Majority outputs 0 (resp. 1); assume n is odd for convenience so there are no ties. Prove that for any function $g_0 : S_0 \to \{0,1\}$, there is a multilinear polynomial $p_0 \in \mathbb{F}_2[X_1, X_2, \ldots, X_n]$ of degree less than $n/2$ such that p_0 and g_0 agree on S_0, and a similar claim holds for any function $g_1 : S_1 \to \{0,1\}$.

   (Hint: Linear algebra. Consider the matrix whose rows are indexed by S_0 and columns by monomials $\prod_{i \in I} X_i$ for $|I| < n/2$, and each entry being the value of the monomial corresponding to that column on the point corresponding to that row.)

(c) Deduce that for every $f : \{0,1\}^n \to \{0,1\}$, there exist polynomials $g, h \in \mathbb{F}_2[X_1, \ldots, X_n]$ of degree $< n/2$ such that $f(x) = g(x)\text{Majority}(x) + h(x)$, $\forall x \in \{0,1\}^n$.

(d) Conclude that if a degree t polynomial $p \in \mathbb{F}_2[X_1, \ldots, X_n]$ computes Majority(x) correctly on $3/4$ of the inputs, then $t \geq \Omega(\sqrt{n})$, and from this deduce the claimed $\exp(\Omega(n^{\frac{1}{2d}}))$ lower bound on size of $\mathsf{AC}^0(\oplus)_d$ circuits computing Majority.

3. **"Majority with margin in AC^0."** The previous exercises have shown that constant depth circuits of polynomial size can't compute Majority. Prove, however, that there is polynomial-sized monotone AC^0 formula of depth 3 that computes a function ClearMajority : $\{0,1\}^n \to \{0,1\}$ satisfying

$$\text{ClearMajority}(x) = \begin{cases} 1 & \text{if } \sum_{i=1}^n x_i \geq 3n/4 \\ 0 & \text{if } \sum_{i=1}^n x_i \leq n/4 \\ \text{don't care} & \text{otherwise} \end{cases}$$

(Hint: Show a probabilistic construction that produces such a circuit w.h.p. Build a depth 3 AND-OR-AND formula of carefully chosen fan-in at each of the 3 levels, with each subtree being an independently sampled formula with the stipulated fan-in structure. Specifically, each bottom AND gate samples $a \log n$ input variables independently, in next layer each OR is fed the output of n^b such AND gates sampled independently, and the top AND gate is fed n^c independent samples of the depth two DNF formula below it (pick constants a, b, c carefully).)

4. **"No not gates please."** Right in the first problem set, you proved that Majority has $O(n)$-sized $O(\log n)$-depth circuits (and therefore also poly(n) sized formulae). However, this construction used NOT gates even though Majority is a monotone function.

(i) Give a simple monotone circuit of $O(n^2)$ size for computing the Majority function on n bits. What is the depth of your circuit?

(Hint: Recall the simple branching program of high width from lecture.)

(ii) Now that we have polynomial-sized monotone circuits for Majority, let's get ambitious and ask about polynomial-sized monotone *formulae* (or equivalently *shallow*, log depth, monotone circuits). Instead of $\vee$, $\wedge$ gates, let's allow ourselves just one kind of gate, MAJ_3, which takes 3 inputs and outputs 1 iff at least 2 of the inputs are 1. Of course, a MAJ_3 gate can be implemented with $O(1)$ many $\vee$ and $\wedge$ gates, so we are free to build a formula with MAJ_3 gates for convenience.

Build the following random monotone formula F with n input bits, *where we assume n is odd for convenience.* The formula will consist of a full ternary tree of depth $c \log n$ for some large enough constant c. Each internal node of the tree will be a MAJ_3 gate. Each of the leaves will be assigned an input variable, uniformly and independently at random. Let us number the levels of the tree from the leaves up, so leaves are at level 0, the first layer of MAJ_3 gates above them are level 1, etc.

Fix an input $x = (x_1, \ldots, x_n) \in \{0,1\}^n$. By symmetry of the construction, all the gates at any given level t must have an identical probability distribution on their outputs. So let us define $p_t = \mathbf{Pr}[\text{gate at level } t \text{ outputs } 1]$. Note that this probability is only over the circuit construction; we treat the input x as fixed for now.

(a) Warm-up: If Majority$(x) = 1$, then $p_0 \geq \frac{1}{2}+\frac{1}{2n}$, and if Majority$(x) = 0$, then $p_0 \leq \frac{1}{2}-\frac{1}{2n}$.

(b) Prove that $p_{t+1} = 3p_t^2(1-p_t) + p_t^3$. What are the fixed points of this recurrence? For later parts, it might help to write an expression for $p_{t+1} - p_t$ in factored form.

(c) Prove that when $\frac{1}{2} + \frac{1}{2n} \leq p_t \leq 3/4$, $p_{t+1} - 1/2 \geq \frac{11}{8}(p_t - 1/2)$. Deduce that for $t_0 = a \log n$ for some large enough a, $p_{t_0} \geq 3/4$.

(d) Prove that when $p_t \geq 3/4$, $(1-p_{t+1}) \leq 3(1-p_t)^2$. Deduce that for $t_1 = b \log n$ for some large enough constant b, $p_{t_0+t_1} \geq 1 - 2^{-(n+1)}$.

(e) Use the above to show that the probability that the random formula F satisfies $F(y) =$ Majority(y) simultaneously for all $y \in \{0,1\}^n$ is at least $1/2$, and therefore Majority can be computed by a monotone formula of depth $O(\log n)$.

5. **"Natural or not?"** Consider the proof in Problem 2 that Majority can't be computed by polysized $\mathsf{AC}^0(\oplus)$ circuits. Recall F_n denotes the set of all Boolean functions on $\{0,1\}^n$; assume n is odd for convenience.

(a) What is a property $C_n \subseteq F_n$ useful against $\mathsf{AC}^0(\oplus)$ that yields the lower bound for Majority? (No need to answer these, but think about: Is the above property large? How about constructive?)

(b) Consider the following property C'_n defined by $C'_n(f_n) = 1$ iff every function $f \in F_n$ can be written as $f(x) = f_n(x)g(x) + h(x)$ where g, h are polynomials of degree $< n/2$. Argue that C'_n is *constructive*, i.e., membership in C'_n can be checked in $2^{O(n)}$ time given the truth table of f_n. (Again no need to answer this, but is C'_n "large"?)

(Hint: Linear algebra. Use the fact that F_n is a $\mathbb{F}_2$-vector space spanned by the 2^n monomial functions $\prod_{i \in I} x_i$ for $I \subseteq \{1, 2, \ldots, n\}$.)

(c) Now consider the following property C^*_n defined by $C^*_n(f_n) = 1$ iff the subspace of functions $f \in F_n$ which can be written as $f(x) = f_n(x)g(x) + h(x)$ for some polynomials g, h of degree $< n/2$ has dimension at least $\frac{3}{4}2^n$.

Argue that C^*_n is *both* large and constructive.

(Hint: For largeness, argue that for every $f_n \in F_n$, $C^*_n(f_n) = 1$ or $C^*_n(f_n \oplus \text{Majority}) = 1$.)

(d) Explain why the proof in Problem 2 is a natural proof.

6. **"No sunflowers please"** The monotone circuit lower bound we showed in class approximated output at each gate by a special form of DNF formula. Let us now develop a method that uses a pair of formulae, one DNF and one CNF, to approximate each gate. For each gate g, we will have a c-CNF formula φ_g (an AND of an arbitrary number of disjunctions of up to c variables each) and a d-DNF formula ψ_g (an OR of an arbitrary number of conjunctions of up to d variables each), for suitable parameters c, d.

Below, your problems appear in italicized font, as they are interspersed with the text.

a) Prove that for every c-CNF formula φ, there is a d-DNF formula ψ on the same variables such that $\psi(a) \leq \varphi(a)$ for every assignment a to input variables, and further there is a collection $\mathcal{C}$ of at most c^{d+1} conjunctions, each of $> d$ distinct variables, such that every b such that $\psi(b) = 0$ and $\varphi(b) = 1$ satisfies at least one of the conjunctions in $\mathcal{C}$.

A similar statement can be shown for rewriting a d-DNF ψ as a c-CNF φ with $\varphi(a) \geq \psi(a)$ $\forall a$, with the extra inputs that the CNF accepts all having the property that they *fail* to satisfy one of at most d^{c+1} disjunctions of $> c$ distinct variables.

(Hint: Think of the standard approach of converting a CNF to a DNF using distributive law. But now we have to drop the DNF terms of width $> d$. Be careful in forming ψ so that more conjunctions have width at most d. It might be helpful to visualize the rewriting process as a tree with edges labeled by variables, and each node v corresponding to a conjunction of the variables on the path from the root to v. We expand the tree from v with the (at most c) edges labeled by variables in the $i+1$'th clause if the distance of v from root is i. But if this clause shares a variable with the root to v path, you can grow just one edge below v....)

You may assume the above for the parts below even if you don't solve it; I've carefully abstracted the approximation so that just the statement is needed to solve the remaining parts.

Armed with the above CNF $\leftrightarrow$ DNF conversion process, we can now define a pair of approximators, one c-CNF φ_C and one d-DNF ψ_C, for a monotone circuit C. For the leaves, we have the exact 1-CNF and 1-DNF consisting of just the variable. Then we have recursively:

- For an AND gate $g = A \wedge B$, we have $\varphi_g = \varphi_A \wedge \varphi_B$ and ψ_g is the d-DNF obtained from φ_g as guaranteed by part (a).
- For an OR gate $g = A \vee B$, we have $\psi_g = \psi_A \vee \psi_B$ and φ_g is the c-CNF obtained from ψ_g as guaranteed by part (a).

Finally, φ_C and ψ_C are the c-CNF and d-DNF at the output gate of the circuit.

We will now use the above approximator-pairs in the approximation method from class to prove an exponential lower bound for a certain natural algebraic monotone function. The function RS is defined as follows. Fix a prime q and let $\mathbb{F}_q$ be the finite field with q elements; let $k < q$ be an integer. The input to RS consists of a $q \times q$ bipartite graph $G = (\mathbb{F}_q, \mathbb{F}_q, E)$ whose vertices on both sides are identified with $\mathbb{F}_q$.

> The function $\mathrm{RS}(G) = 1$ iff there exists a univariate polynomial $p \in \mathbb{F}_q[X]$ of degree $\leq k$ such that $(a, p(a)) \in E$ for all $a \in \mathbb{F}_q$. In other words, the graph G contains the edges corresponding to the evaluations of some low-degree polynomial on $\mathbb{F}_q$.

We now define positive and negative test cases as follows.

- The positive test graphs are what one would expect. For each degree of the q^{k+1} polynomials p of degree $\leq k$, we define the test graph G_p with edge set $E_p = \{(a, p(a)) \mid a \in \mathbb{F}_q\}$.
- The negative test graphs are defined randomly, by including each edge independently with probability $1 - \epsilon$ for $\epsilon = (2k \ln q)/q$. (Note this construction might result in all possible $q \times q$ bipartite graphs, but you'll prove below that it is unlikely to yield a positive test case.)

b) Prove that the probability that a negative test graph G, sampled as above, satisfies $\mathrm{RS}(G) = 1$ *is at most* $q^{-\Omega(k)}$.

Suppose C is a monotone circuit with fan-in two $\vee$ and $\wedge$ gates that computes RS. We will compute the CNF/DNF approximator pairs with parameters defined below:

$$d = k \quad \text{and} \quad c = \lfloor q^{2/3}/2 \rfloor \ .$$

c) Consider the c-CNF approximator φ_C. Prove that it is either identically 1, or fails to accept at least half the positive test graphs.

As in class, we say that the CNF/DNF approximator pair *introduces* an error on a input graph G at gate g, if the approximator-pairs at the gates feeding g are both correct on G, but the output approximator (either one in the pair) is incorrect on G. Note that at an AND (resp. OR) gate, only the DNF (resp. CNF) approximator can introduce an error.

d) At an AND gate g, the d-DNF ψ_g introduces an error for at most c^{d+1} positive test graphs.

e) At an OR gate g, the probability that the c-CNF φ_g introduces an error for a random negative test graph is at most $(d\epsilon)^{c+1}$.

f) Using parts b,c,d, and e, argue that the size of C must be at least $q^{\Omega(k)}$ when $k \leq q^{1/4}$.