15-855: An Introduction to Computational Complexity Theory Fall 2015
Problem Set #4
Due on **November 10, 2015** (in class, or by email to yuzhao1@cs.cmu.edu).

Instructions: Same as for problem set 1.

Solve the first 5 problems. The last one on Discrete Fourier Transform is not to be turned in.

1. **"Non-uniform complexity relationships."** For a Boolean function $f : \{0,1\}^n \to \{0,1\}$, let (i) $C(f)$ be the smallest number of gates in a circuit with $(\wedge, \vee, \neg)$ gates (of fan-in at most 2) that computes f; (ii) $BP(f)$ denote the smallest number of nodes in a branching program computing f; and (iii) $L(f)$ denote the minimum number of a leaves in a $(\wedge, \vee, \neg)$-formula (again with fan-in 2) computing f.

   (a) Prove the inequalities mentioned in class: $C(f) \le O(BP(f)) \le O(L(f))$.

   (b) Prove that a family $\mathcal{F}$ of Boolean functions $\{f_n : \{0,1\}^n \to \{0,1\}\}_{n \ge 1}$ has polynomial size formulas, i.e., $L(f_n) \le cn^c$ for all n for some absolute constant $c < \infty$, if and only if $\mathcal{F}$ is in non-uniform NC^1.

   (Hint: The main thing to show is that formulas can be rebalanced to logarithmic depth in their size, as mentioned without proof in class.)

2. **"Decision trees: Deterministic, Nondeterministic, and Unambiguous."** For a Boolean function $f : \{0,1\}^n \to \{0,1\}$, define the following decision tree complexities:

   - $\mathsf{P}^{\mathsf{dt}}(f) =$ smallest depth of a decision tree computing f.
   - $\mathsf{NP}^{\mathsf{dt}}(f) =$ the minimum cost of a nondeterministic decision tree for f, where a nondeterministic decision tree is a collection of 1-certificates, that is partial assignments to variables of f that force the function to be 1, and its cost is the maximum number of variables fixed by such a partial assignment. In other words, a nondeterministic decision tree is is just a DNF formula, and its cost is the maximum width of its terms, that is the width of the DNF.
   - $\mathsf{UP}^{\mathsf{dt}}(f) =$ the minimum cost of an unambiguous decision tree for f, where a nondeterministic decision tree is unambiguous if for each input, there is at most one accepting certificate. In other words, an unambiguous decision tree is a DNF formula where for each input there is at most one term in the DNF formula that evaluates to true.

   (a) Prove that $\mathsf{NP}^{\mathsf{dt}}(f) \le \mathsf{UP}^{\mathsf{dt}}(f) \le \mathsf{P}^{\mathsf{dt}}(f)$ for every Boolean function f.

   (b) Give an example of $f : \{0,1\}^n \to \{0,1\}$ with $\mathsf{P}^{\mathsf{dt}}(f) = n$ and $\mathsf{NP}^{\mathsf{dt}}(f) = 1$.

   (c) Give an example of $f : \{0,1\}^n \to \{0,1\}$ with

   $$\mathsf{P}^{\mathsf{dt}}(f) \ge \Omega(n) \quad \text{and} \quad \max\{\mathsf{NP}^{\mathsf{dt}}(f), \mathsf{NP}^{\mathsf{dt}}(\neg f)\} \le O(\sqrt{n}) \tag{1}$$

   where $\neg f$ stands for the negation of f. What is the asymptotic value of $\mathsf{UP}^{\mathsf{dt}}(f)$ for your example function f?

(Can you make the quadratic gap in (1) any larger? No need to turn in the answer to this question, but it is related to one of your midterm problems. Another related fact is that $\mathsf{P}^{\mathsf{dt}}(f) \le \mathsf{UP}^{\mathsf{dt}}(f)^2$, and the exercise below gives an example with nearly quadratic gap between deterministic and unambiguous decision tree complexity.)

(d) (Warmup) Let $n = m^2$ and consider the function $f : \{0,1\}^{m\times m} \to \{0,1\}$ defined on Boolean matrices $M \in \{0,1\}^{m\times m}$ such that $f(M) = 1$ iff M has a *unique* all-1 column. Argue that

$$\mathsf{NP}^{\mathsf{dt}}(f) \le 2m - 1 \qquad \text{and} \qquad \mathsf{P}^{\mathsf{dt}}(f) = m^2 \ .$$

(e) Modify the previous function, with a view towards bounding $\mathsf{UP}^{\mathsf{dt}}$, as follows. We will give a function $g : \Sigma^{m\times m} \to \{0,1\}$ whose input alphabet is $\Sigma = \{0,1\} \times P$ where $P = ([m] \times [m]) \cup \{\perp\}$ and $[m]$ denotes $\{1, 2, \ldots, m\}$. Note that a decision tree for g will thus read values from Σ for each variable and will be $|\Sigma|$-ary instead of binary (its decision tree complexities can be defined analogously to the Boolean input case, though for nondeterministic and unambiguous case we can't express it as a Boolean DNF formula). [1]

The set $P = ([m] \times [m]) \cup \{\perp\}$ is to be thought of a set of *pointer values*, where we interpret an entry $M_{ij} = (m_{ij}, p_{ij}) \in \Sigma$ in the (i,j)'th *cell* of the matrix as *pointing* to another entry $M_{p_{ij}}$ if $p_{ij} \neq \perp$, and refer to $m_{ij} \in \{0,1\}$ as the *value* in the (i,j)'th cell. If $p_{ij} = \perp$ then we have a null pointer.

Finally, we define the function $g : \Sigma^{m\times m} \to \{0,1\}$ as follows: $g(M) = 1$ if and only if:

- M has a column where all but one cell contains the entry $(1, \perp)$, the remaining cell contains $(1, p)$ for $p \neq \perp$; and
- Following the pointer from p successively for $m - 1$ steps leads us to one cell in each of the remaining $(m-1)$ columns, where all those cells have value 0 and the last pointer value is $\perp$ (i.e., we have a chain of cells with entries $(1, p) \to (0, p_1) \to (0, p_2) \to \cdots \to (0, p_{m-2}) \to (0, \perp)$ where $p_1, \ldots, p_{m-2} \neq \perp$, and no two of these cells lie in the same column).[2]

For the above function, prove the quadratic separation

$$\mathsf{P}^{\mathsf{dt}}(g) = m^2 \quad \text{and} \quad \mathsf{UP}^{\mathsf{dt}}(g) \le 2m - 1 \ .$$

3. **"Exact formula size for parity."** The parity function on n inputs is

$$\bigoplus_n (x_1, x_2, \ldots, x_n) = x_1 \oplus x_2 \oplus \cdots \oplus x_n$$

i.e., it is 1 if and only if there are an odd number of 1's among its n inputs.

(a) Show that $L(\bigoplus_n) \le n^2$ when n is a power of 2.

(b) A *formal complexity measure* FC is a function mapping Boolean functions on n variables to the natural numbers, and satisfying the following properties:

[1] One can get a near-quadratic gap, with some logarithmic losses, for functions with a Boolean input alphabet by considering $g \circ h^n$ where $h : \{0,1\}^{\lceil \log |\Sigma| \rceil} \to \Sigma$ is some onto map, but for simplicitly let's ignore this aspect.

[2] I really should have drawn a picture for this, so please ask if something isn't clear, either about the function, or what $\mathsf{UP}^{\mathsf{dt}}(g)$ means for non-Boolean inputs.

i. $\mathsf{FC}(x_i) = 1$ for $1 \leq i \leq n$

ii. $\mathsf{FC}(f) = \mathsf{FC}(\neg f)$ for all f

iii. $\mathsf{FC}(f \vee g) \leq \mathsf{FC}(f) + \mathsf{FC}(g)$ for all f, g

Show that for *every* formal complexity measure FC we have $\mathsf{FC}(f) \leq L(f)$.

(c) For subsets A and B of $\{0,1\}^n$, define

$$\begin{aligned} H(A,B) &= \{(a,b) : a \in A, b \in B, a \text{ and } b \text{ differ in exactly 1 coordinate}\} \\ K(f) &= \max_{A \subseteq f^{-1}(1),\, B \subseteq f^{-1}(0)} \frac{|H(A,B)|^2}{|A||B|}. \end{aligned}$$

Show that K is a formal complexity measure.

(Hint: To prove property (iii), which is the main part, partition A into $A_f \subseteq f^{-1}(1)$ and $A_g \subseteq g^{-1}(1)$ for the subsets A, B that maximize the expression defining $K(f \vee g)$.)

(d) Show that $L(\bigoplus_n) \geq n^2$.

4. **"Neciporuk again."** The indirect access function (IDA) is defined as follows. For an integer $\ell \geq 1$, let $m = 2^{2^\ell}$, $k = 2^\ell - \ell$, and $n = 2m + k$. For $x = (x_0, x_1, \ldots, x_{m-1})$, $y = (y_0, y_1, \ldots, y_{m-1})$ and $b = (b_0, b_1, \ldots, b_{k-1})$,

$$\mathrm{IDA}_n(x, y, b) = y_{|t|}$$

for $t = (x_{|b| \log m}, x_{|b| \log m + 1}, \cdots, x_{|b| \log m + \log m - 1})$ where $|z|$ denotes the binary number represented by the bit-string $z = (z_0, \ldots, z_p)$.

Prove that $L_{B_2}(\mathrm{IDA}_n) \geq \Omega(n^2 / \log n)$, where $L_{B_2}(f)$ is the minimum number of leaves in a formula over the full binary basis (of all 16 functions on two inputs) that computes f. Note that this lower bound is a $\log n$ factor better than the lower bound for branching programs.

5. **"Decision trees and Fourier spectrum."** Suppose $f : \{0,1\}^n \to \{0,1\}$ is computed by a decision tree T with depth d and M leaves. Let $\widehat{f}(\alpha)$ be the Fourier coefficient of f in the notation of Problem 6, and let sparsity(f) be the number of nonzero $\widehat{f}(\alpha)$ among $\alpha \in \{0,1\}^n$. Prove that

(a) $\frac{1}{2} \log(\text{sparsity}(f)) \leq d$,

(b) $\sum_{\alpha \in \{0,1\}^n} |\widehat{f}(\alpha)| \leq M$.

(Hint: Express $f = \sum_v f_v$ as the sum of over leaves v of T of functions f_v such that $f_v(x)$ equals the value at the leaf v if the computation of T on input x ends up at leaf v, and 0 otherwise.) Then compute the Fourier spectrum of each f_v.)

6. **"Fourier basics."** (*This question need* **not** *be turned in. But if you aren't familiar with this, I recommend you work it out anyway. You can use any of these facts for Problem 5 above.*) Let $\mathcal{F} = \{f \mid f : \{0,1\}^n \to \mathbb{R}\}$ be the space of real-valued functions on $\{0,1\}^n$. For two functions $f, g \in \mathcal{F}$ define their inner product

$$\langle f, g \rangle = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} f(x) g(x) \; . \tag{2}$$

For $\alpha \in \{0,1\}^n$, define the function $\chi_\alpha : \{0,1\}^n \to \mathbb{R}$ by $\chi_\alpha(x) = (-1)^{\alpha \cdot x}$ (here $\alpha \cdot x$ denotes the dot product of the vectors α and x over the reals).

a) Prove that the functions $\{\chi_\alpha\}_{\alpha \in \{0,1\}^n}$ form an *orthonormal* basis for the space $\mathcal{F}$ with respect to the inner product (2) defined above.

b) Conclude that every $f \in \mathcal{F}$ has a *unique* representation as

$$f(x) = \sum_{\alpha \in \{0,1\}^n} \widehat{f}(\alpha)\ \chi_\alpha(x)$$

for real coefficients $\widehat{f}(\alpha)$ given by $\widehat{f}(\alpha) = \langle f, \chi_\alpha \rangle$.

c) Prove that for $f, g \in \mathcal{F}$, the following identity holds:

$$\langle f, g \rangle = \sum_{\alpha \in \{0,1\}^n} \widehat{f}(\alpha)\ \widehat{g}(\alpha)\ .$$

Deduce that for a function $f : \{0,1\}^n \to [-1,1]$, $\sum_\alpha \widehat{f}(\alpha)^2 \leq 1$.

d) Suppose $f \in \mathcal{F}$ is invariant under translations by a vector $h \in \{0,1\}^n$, i.e., $f(x+h) = f(x)$ for all x (here $x + h$ is computed by component-wise addition modulo 2).

Prove that $\widehat{f}(\alpha) = 0$ whenever $\alpha \cdot h = 1$ modulo 2.