CS 85/185
Fall 2003
Lower Bounds in Computer Science

Homework 4
Due Dec 2, 2003

Amit Chakrabarti
Department of Computer Science
Dartmouth College

Every solution fits within a page, so please don't use more space than that. And as usual, please think carefully about how you are going to organise your proofs *before* you begin writing.

1. Suppose Alice has an n-bit string $x = x_0 x_1 \ldots x_{n-1}$ and Bob has an n-bit string $y = y_0 y_1 \ldots y_{n-1}$ with the *promise* that $x \neq y$. Their goal is to *agree* on any one index i such that $x_i \neq y_i$; we call this the **distinguishing bit problem**. Note that there may be several acceptable answers for the problem! A trivial protocol for this problem works as follows: Alice sends Bob the whole string x and Bob replies with a suitable index i. This uses $n + \lceil \lg n \rceil$ bits of communication.

   Describe a protocol which uses only $n + \lg^* n$ bits. Recall that $\lg^* n$ is informally defined as the number of times you need to "apply the $\lg$ function" before n reduces to 1.

2. The "greater than" function $\mathsf{GT}_n : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ is defined as follows. We interpret the input strings $x, y \in \{0,1\}^n$ as integers between 0 and $2^n - 1$ and let

$$\mathsf{GT}_n(x,y) \quad = \quad \begin{cases} 1, & \text{if } x > y \\ 0, & \text{otherwise}\,. \end{cases}$$

   2.1. Show that $\mathrm{D}(\mathsf{GT}_n) \geq n$. What lower bound, if any, does this imply for $\mathrm{R}(\mathsf{GT}_n)$ and why?

   2.2. Recall the randomized protocol we described in class for the equality problem. Show that $\mathrm{R}(\mathsf{GT}_n) = O(\log^2 n)$ by repeated application of this protocol. **Alert:** Remember that the equality protocol makes a mistake with some probability. When you apply the protocol multiple times, the error can add up. Make sure you analyze that.

3. Improve the construction from Problem 2.2 and show that $\mathrm{R}(\mathsf{GT}_n) = O(\log n \log\log n)$.

   Hint: First consider $\mathrm{R}^{\mathrm{pub}}(\mathsf{GT}_n)$.

   For extra credit, prove that in fact $\mathrm{R}(\mathsf{GT}_n) = O(\log n)$. Warning: not easy!

4. This problem uses what I call the **critical set lemma** which we will eventually prove in class. For now, you can solve the problem assuming the lemma to be true. The lemma goes like this: let X and Y be *disjoint* subsets of $\{0,1\}^n$. Consider the distinguishing bit problem where Alice is given $x \in X$ and Bob is given $y \in Y$ (note that the disjointness condition ensures $x \neq y$). The critical set lemma says that the deterministic communication complexity of the problem is at least $\log \frac{|C|^2}{|X| \cdot |Y|}$, where

$$C \quad = \quad \{(x,y) \in X \times Y : x \text{ and } y \text{ differ in exactly one bit position}\}$$

   is the critical set of the problem.

   Let D_n be the deterministic communication complexity of the distinguishing bit problem where Alice is given an odd parity n-bit string x and Bob is given an even parity n-bit string y.

   4.1. Prove that $D_n \leq 2\lceil \lg n \rceil$. Hint: Binary search.

   4.2. Using the critical set lemma, prove that $D_n \geq 2 \lg n$. Conclude that $D_n = 2\lceil \lg n \rceil$.

[See other side]

5. Consider the following communication problem involving *three* players: Alice, Bob, and Carol. Each of Alice and Bob has the n-bit string $x = x_0 x_1 \dots x_{n-1}$ (note: they both have this same string). Moreover, Alice has an index $i \in \{0, 1, \dots, n-1\}$ and Bob has an index $j \in \{0, 1, \dots, n-1\}$. Carol only knows n, i, and j and needs to find out the bit $x_{(i+j) \bmod n}$.

   The twist in the model is that each of Alice and Bob can send *one* message to Carol and then Carol must compute the answer. No other messages may be exchanged; in particular Alice and Bob cannot talk to each other. Everything is deterministic, so Carol must always get it right. There is a trivial protocol in which Alice sends n bits (the entire string x) and Bob sends 0 bits.

   Find a protocol in which Alice sends at most $\lceil n/2 \rceil$ bits and Bob sends 1 bit.

   Important Research Question: Just so you know why one bothers about this problem, it is a big deal to figure out if there is a protocol in which Alice sends $o(n)$ bits and Bob sends just a few bits, say $O(\log n)$. This question has important implications for circuit complexity.