COS 598a

February 26, 2001

Assignment 3

Homeworks We use the notation from the Appendix.

Exercise 1 Prove that any randomized one-round protocol to compute f must satisfy the inequality $\ell_A \cdot \ell_B \geq \Omega(D_0(f))$.

Hint: Let $x \in \{0,1\}^n$. For each possible message $\phi \in \{0,1\}^{\ell_B}$ generated by B, let $F(x,\phi)$ be the probability that $Q_C(w,\phi) = 1$ for a random message $w \in \{0,1\}^{\ell_A}$ generated by A (for input x). Let $t = 1000\ell_B$. Prove that for each $x \in \{0,1\}^n$, there exists a set $W_x = \{w_1, w_2, \cdots, w_t\}$, where $w_i \in \{0,1\}^{\ell_A}$, such that for every $\phi \in \{0,1\}^{\ell_B}$,

$$|\frac{1}{t} \sum_{1 \leq i \leq t} Q_C(w,\phi) - F(x,\phi)| < \frac{1}{10}.$$

Infer that $N_A(f) \leq 1000\ell_A \cdot \ell_B$.

Exercise 2 Let $n > 0$. For any $x, y \in \{0,1\}^n$, let $d_H(x,y)$ denote the Hamming distance between x, y. Let $f(x,y) = 1$ if $d_H(x,y) \leq 1$ and 0 otherwise. Determine $R(f)$ for large n as best as you can.

Exercise 3 Consider the communication problem where initially A holds a boolean array $x[0:n-1]$ and integer $0 \leq i < n$, B holds the same boolean array $x[0:n-1]$ and integer $0 \leq j < n$, and processor C holds the integers n, i, j. Let $f(x,i,j) = x[i+j \bmod n]$. Processors A and B send strings to processor C in one round. Processor C needs to be able to compute the value of $f(x,i,j)$. Give a deterministic one-round protocol in which A sends 1 bit, and B sends $\lceil n/2 \rceil$ bits.

Research Problems

Problem 1 For $x = (x_1, x_2, \cdots, x_m), y = (y_1, y_2, \cdots, y_m)$ where $x_i \in \{0,1\}^n, y_i \in \{0,1\}^n$ for all i, define $f(x,y) = 1$ if there are at least $m/2$ i such that $x_i = y_i$, and otherwise 0. What is $R(f)$? Clearly, $R(f) \leq O(m\sqrt{n})$. Also, $R(f) \geq \Omega(\sqrt{mn})$ by Theorem 1.

Problem 2 More generally, can you find examples of f with interesting $R(f)$? By that we mean with upper bounds not obtained from Theorem 2 (in the Appendix), or lower bounds not obtained from Theorem 1.

Problem 3 For the communication problem in Exercise 3 above, if Processor B is allowed to send only $n/3$ bits, how many bits does processor A have to send? It is a question of great interest whether there is a protocol with B sending $o(n)$ bits and A sending only a small number of bits. For example, if A can send only $\log n$ bits, can B send just $o(n)$ bits?

Notes The results on the topic discussed in class and in the Appendix are from Ambainis[6], Newman and Szegedy[7] and Babai and Kimmel[8]. For information on the topic discussed in Exercise 3 and Research Problem 3, see Pudlak, Rodel and Sgall[9]. The significance of this latter topic to circuit complexity was first discovered by Nisan and Wigderson[10]. The outstanding open conjecture is that, for some fixed $\epsilon > 0$, there is no protocol in which A sends $O(n^\epsilon)$ bits and B sends $O(\frac{n}{\log\log n})$ bits. A positive proof of this conjecture would resolve a long-standing open problem in circuit complexity.

Appendix: Simultaneous Messages Model

Let $f(x, y)$ be a boolean-valued function. Processors A, B hold respectively $x, y \in \{0,1\}^n$, and a third processor C, called the *referee*, holds initially nothing. A *one-round protocol* for computing f consists of three algorithms Q_A, Q_B, Q_C, such that if A, B send respectively two strings $a \in \{0,1\}^{\ell_A}, b \in \{0,1\}^{\ell_B}$ (a is calculated by A using Q_A on input x, and b by B using Q_B on y) to C who then proceeds to compute a result c using Q_C on inputs a, b. The value of $\max\{\ell_A, \ell_B\}$ is the *communication cost* of the protocol. In the *deterministic model*, Q_A, Q_B, Q_C are all deterministic, and c is required to be equal to $f(x, y)$. In the *randomized model*, Q_A, Q_B, Q_C are all randomized algorithms, and for every input x, y, c is required to be equal to $f(x, y)$ with probability $\geq 99/100$. (The choice of 99/100 is arbitrary. One can use any constant $1/2 < \lambda < 1$ instead.) Let $D_0(f)$ be the minimum cost of any protocol in the deterministic model; let $R(f)$ be the minimum cost of any protocol in the randomized model. Let $N_A(f)$ be the number of distinct rows of the $2^n \times 2^n$ matrix $(f(x, y))$, and $N_A(f)$ be the number of distinct columns. Clearly $D_0(f) = \lceil \log_2 \max\{N_A(f), N_B(f)\} \rceil$.

Let $EQUAL(x, y) = 1$ if $x = y$ and 0 otherwise. Then $D_0(EQUAL) = n$, and $R(EQUAL) = \theta(\sqrt{n})$. The deterministic case is obvious, so we concentrate on the randomized case. We discuss two general theorems from which the result follows easily.

Theorem 1 Any randomized one-round protocol to compute f must satisfy the inequality $\ell_A \cdot \ell_B \geq \Omega(D_0(f))$.

The proof of Theorem 1 is left as an exercise in Assignment 3.

Consider a new randomized model in which the random string (of any length) is shared by both A and B. Call this the *public-coin* randomized model, and $R'(f)$ the corresponding complexity. (The previous one is then called the *private-coin* randomized model, if we need to distinguish them.) Note that $R'(EQUAL) = O(1)$. Observe that if $x \neq y$, then for a random r, the inner product $r \cdot (x - y) \bmod 2$ has exactly probability $1/2$ to be a 1. In contrast, if $x = y$, clearly the inner product is always 0. This immediately gives an $O(1)$ public-coin protocol for computing EQUAL using a public random string r. (Strictly speaking, you need to use $\lceil \log_2 100 \rceil$ random r's to get error probability below 1/100.)

Theorem 2 $R(f) = O(R'(f)\sqrt{n})$.

The proof of Theorem 2 relies on the following observation. Let $2/3 < \lambda \leq 1$ and $\epsilon > 0$ be constants. Let $g : X \to \{0,1\}$, $G : X \times \{0,1\}^* \to \{0,1\}$, and ρ a probability distribution on $\{0,1\}^*$. Assume that for any x, if we take a random r distributed according to ρ, then $G(x,r) = g(x)$ with probability at least λ. That is, assume that g is randomly computable using G.

Lemma 1 There exists $m = O(\log |X|)$ binary strings $r_1, r_2, \cdots, r_m$ such that, for every $x \in X$, at least $\lambda - \epsilon$ fraction of the r_i $(1 \leq i \leq m)$ satisfy $G(x, r_i) = g(x)$.

The proof of Lemma 1 is elementary. Let $x \in X$ be any input. Generate m random r_i according to ρ, then we expect that at least about λ fraction of the m strings satisfy $G(x, r_i) = g(x)$. In fact, the probability of having less than $(\lambda - \epsilon)m$ such r_i is $2^{-\Omega(m)}$. Since there are $|X|$ possible x, the probability of having less than $(\lambda - \epsilon)m$ r_i satisfying $G(x, r_i) = g(x)$ for *any* x is less than $2^{-\Omega(m)}|X| < 1$. This proves the Lemma.

To prove Theorem 2, we can assume without loss of generality that Q_C is deterministic, and that the probability for the public-coin protocol to succeed is at least 99.5/100. By Lemma 1, there exist $m = O(\log(2^n \cdot 2^n)) = O(n)$ strings $r_1, r_2, \cdots, r_m$ such that the following is true: for any pair (x, y), for at least 99/100 fraction of the $1 \leq i \leq m$, $Q_C(a_i, b_i) = f(x, y)$ (where $a_i = Q_A(x, r_i)$ and $b_i = Q_B(y, r_i)$). We now give a private-coin protocol. Arrange the $m = O(n)$ indices $1 \leq i \leq m$ as an $O(\sqrt{n}) \times O(\sqrt{n})$ array. Processor A picks a random column j, computes $a_{i,j} = Q_A(x, r_{i,j})$ for all $1 \leq i \leq O(\sqrt{n})$ and sends them and j all to C. Processor B picks a random row i, computes $b_{i,j} = Q_B(y, r_{i,j})$ for all $1 \leq j \leq O(\sqrt{n})$ and sends them and i all to C. Processor C decides $Q_C(a_{i,j}, b_{i,j})$ as the value. By the above discussions, C has at least probability 99/100 of outputing the correct value of $f(x, y)$. Note that the communication cost in this protocol is $O(R'(f)\sqrt{n})$. This proves Theorem 2.