CS277: Concrete Complexity	**Fall 1996**

Lecture 19: November 5, 1996

Lecturer: Uri Zwick — *Scribe: Franklin Cho*

19.1 Communication Complexity

Suppose Alice has x and Bob has y, where x and y are both n-bit vectors. Let us consider the communication complexity of the following problems:

EQUALITY: Is $x = y$? (Both parties must know the answer.)

UNIVERSAL RELATION: Given that $x \neq y$, find i such that $x_i \neq y_i$.

Notice that given a function $f : \{0,1\}^n \to \{0,1\}$, we can define a relation R_f: promise that $f(x) = 0$ and $f(y) = 1$. Find i such that $x_i \neq y_i$. Computing the Universal Relation U_n over a communcation line would compute R_f. Therefore,

$$D(f) = C(R_f) \leq C(U_n)$$

DISJOINTNESS: Is $\bar{x} \cap \bar{y} = \phi$? (Here, $\bar{x}$ denotes the subset of $\{1, \ldots, n\}$, whose characteristic vector is x.) Or, equivalently, is $x_i = 0 \vee y_i = 0$ for every i?

19.1.1 Upper and Lower Bounds

A function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ can be represented by a $2^n \times 2^n$ matrix. For every $2^n \times 2^n$ matrix M, it is easy to see that $C(M) \leq n + 1$, since Alice can send Bob her word, and Bob can send the result back to Alice. Therefore,

$$C(EQ_n) \leq n + 1$$
$$C(DISJ_n) \leq n + 1$$

We will later show that these upper bounds are tight. Also, by a similar argument,

$$C(U_n) \leq n + \lceil \log n \rceil$$

Exercise:

Construct a better protocol for the Universal Relation.

■

Notice that $C(EQ_n) \leq C(U_n) + 2$. (After U_n, Alice sends Bob x_i, and Bob sends Alice whether $x_i = y_i$.) Therefore, $C(U_n) \geq C(EQ_n) - 2 = (n+1) - 2 \geq n - 1$.

$$C(U_n) \geq n - 1$$

Theorem 19.1 *(The rank lower bound) If M is a non-constant matrix (not identically 0 or 1), then*

$$C(M) \geq \log rk(M) + 1$$

where $rk(M)$ is the rank (the number of independent columns) of the matrix M.

Proof: (By induction on rank)

If $rk(M) = 0$, then M is identically 0, which violates our assumptions about M.

If $rk(M) = 1$, then M is either identically 1, or it contains both 1's and 0's. If M is identically 1, this violates our assumptions about M. If M contains both 1's and 0's, we do not know in advance what the output of the function is. Therefore, at least one bit must be sent, and $C(M) \geq 1$.

If $rk(M) = 2$, divide M into two submatrices M_0 and M_1. Here, M_0 is the collection of rows of M for which the first bit sent is 0. M_1 is the collection of rows of M for which the first bit sent is 1. We continue to subdivide one of these matrices until we have a constant matrix ($rank$ = 0 or 1). Since $rk(M) \leq rk(M_0) + rk(M_1)$, the shortest communication sequence would produce $rk(M_0) = rk(M_1) = 1$. Since $C(M_0) \geq 1$ and $C(M_1) \geq 1$, $C(M) \geq 1 + 1 = 2$.

If $rk(M) > 2$, we divide M into two submatrices, M_0 and M_1, based on the first bit sent.

Since

$$rk(M) \leq rk(M_0) + rk(M_1),$$

it follows that

$$\exists i \text{ st } rk(M_i) \geq \frac{1}{2} rk(M).$$

Then, perform the communication protocol on the submatrix M_i. By the induction hypothesis,

$$C(M) = 1 + C(M_i) \geq 1 + (\log rk(M_i) + 1)$$

If we assume that $rk(M_i) \geq \frac{1}{2} rk(M)$, then,

$$C(M) \geq \log rk(M) + 1$$

∎

Corollary

$C(EQ_n) = n + 1$ since the rank of the identity matrix is 2^n.

$C(DISJ_n) = n + 1$.

Proof:

We can construct the $DISJ_n$ matrix by the following recursive definition.

$$DISJ_1 = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$$

$$DISJ_n = \begin{pmatrix} DISJ_{n-1} & DISJ_{n-1} \\ DISJ_{n-1} & 0 \end{pmatrix}$$

To prove that $rk(DISJ_n) = 2^n$, we will use induction on n. If $n = 1$, the proof is trivial. In general case, we will prove that only a trivial combination of the rows produces a zero vector. In order to produce a zero sum vector, the left halves of the rows must combine to produce a zero sum vector. This reduces to $DISJ_{n-1}$ case, and by the induction hypothesis, the proof is finished.

∎

19.2 Randomized Communication Complexity

We define a *randomized communication complexity* as follows. Alice and Bob are allowed to flip coins. The coins can be public or private. Alice and Bob may get a wrong result with *prob* $\leq \varepsilon$. The complexity of the a protocol is defined to be the maximal number of bits sent over all inputs and all random bits.

- $C_\varepsilon(M)$ = The complexity of the best randomized protocol for M that errs with *prob* $\leq \varepsilon$.

$$C^{public}_{\frac{1}{4}}(EQ_n) = O(1)$$

Proof:

Pick $a_1, a_2, \ldots, a_n \in \{0,1\}$ at random. Alice sends $a \cdot x = \sum a_i \cdot x_i \pmod 2$. Bob checks whether $a \cdot x = a \cdot y$ and send the answer. The error rate $\varepsilon = 1/2$. To get $\varepsilon \leq 1/4$, repeat the protocol.

■

$$C^{private}_{\frac{1}{4}}(EQ_n) = O(\log n)$$

Proof:

Alice picks a random prime number p between $1, \ldots, n^2$. She sends p ($2 \log n$ bits) and $x \pmod p$ ($2 \log n$ bits) to Bob. Bob checks whether $x \pmod p = y \pmod p$, and sends the answer to Alice.

If $x = y$ the result of the protocol is always correct. If $x \neq y$, the protocol can err. The protocol errs when Alice picks a prime p such that p divides $|x - y|$. Now, $|x - y| < 2^n$, and therefore the number of primes dividing $|x - y|$ is no more than $\log 2^n = n$. We are choosing a prime number with equal probability from a collection of size $n^2/\log n^2$. Therefore, we get the following estimate:

$$\varepsilon \leq \frac{n}{n^2/\log n^2} = \frac{2\log n}{n} \leq \frac{1}{4}$$

■

$C^{public}_{\frac{1}{4}}(DISJ_n) = \Omega(n)$. This lower bound was proved by Kalyanasundaram and Schnitger. A considerably simpler proof was found by Razborov.

Given a monotone function $f : \{0,1\}^n \to \{0,1\}$, we can define a relation $R_f^M = \{(x, y, i) : f(x) = 0; f(y) = 1, x_i < y_i\}$.

Theorem 19.2 $C(R_f^M) = D_M(f)$.

Exercise:

Prove the theorem.

■

CS277: Concrete Complexity **Fall 1996**

Lecture 20: November 7, 1996

Lecturer: Uri Zwick *Scribe: Franklin Cho*

20.1 Communication Complexity of Monotone Functions

Communication problem of monotone functions is defined as follows: Given a function $f : \{0,1\}^n \to \{0,1\}$, Alice gets x such that $f(x) = 0$ and Bob gets y such that $f(y) = 1$. Find i such that $x_i < y_i$.

Monotone function $MATCH_n(x)$ returns 1 iff the graph described by x contains a matching of size m, where $x = \{x_{ij}\}, 1 \leq i < j \leq n$ and $n = 3m$.

20.1.1 Using Communication Complexity to obtain a Lower Bound on Depth of a Monotone Function

We define following communication problems:

$MATCH_n$: Bob gets a maching p of size m. Alice gets the complement of a clique c of size $2m+1$. (Therefore, Alice gets a maximal set that does not contain a matching of size m. $MATCH_n(m) = 1$ and $MATCH_n(c) = 0$.) Equivalently, Alice gets a set q of $m-1$ vertices. Find an edge e such that $e \in p$ and $e \cap q = \phi$.

$MATCH'_n$: Bob gets a matching p and Alice gets a set of vertices $\hat{q}$ of size m. Is there e such that $e \in p$ and $e \cap \hat{q} = \phi$?

$DIST_n$: Alice gets $x \in \{0,1,2\}$. Bob gets $y \in \{0,1,2\}$. Is $x_i \neq y_i$ for every $1 \leq i \leq n$?

$DISJ_n$: Alice gets $x \in \{0,1\}^n$. Bob gets $y \in \{0,1\}^n$. Is $x_i = 0 \vee y_i = 0$ for every $1 \leq i \leq n$?

Through a series of reductions, we will show that

$$\Omega(n) = C_{\frac{1}{3}}(DISJ_m) \leq C_{\frac{1}{3}}(DIST_m) \leq C_{\frac{1}{3}}(MATCH'_n) \leq C(MATCH_n) \leq D_M(MATCH_n)$$

Proof:

20.1.1.1 Reduction from $MATCH'_n$ to $MATCH_n$

Assume that $PROT$ is a deterministic protocol for $MATCH_n$. Consider a probabilistic protocol $PROT''$ for $MATCH_n$:

$PROT''$: Suppose $\{e_1, e_2, \ldots, e_i\}$ is the set of edges in p which do not intersect q. Alice and Bob flip coins publicly and choose a permutation π on the vertices of the graph. Then, execute $PROT$ on πp and πq. It is evident that $PROT''$ returns each edge from $\{e_1, e_2, \ldots, e_i\}$ with equal probability.

We construct a probabilistic protocol $PROT'$ for $MATCH'_n$ as follows.

$PROT'$: Given a matching p and a set $\hat{q}$, Alice chooses a random element $v \in \hat{q}$ and defines $q = \hat{q} \backslash \{v\}$.

Alice and Bob run $PROT''$ on p and q. Eventually, Alice and Bob agree on an edge e such that $e \in p$ and $e \cap q = \phi$. Alice checks whether $v \in e$ and reports to Bob.

If $v \notin e$ then $e \cap \hat{q} = \phi$ and Alice and Bob know that the answer is 1. Otherwise, they gamble on 0.

Suppose set B contains all the edges in p with exactly one vertex in $\hat{q}$, and set C contains all the edges in p which contain no vertex in $\hat{q}$. The gamble will be wrong if $C \neq \phi$.

To produce an error, v must be contained in B, C must contain at least one edge, and $PROT$ returns an edge in B. The probability of picking an edge in B is $\leq 1/2$. So, the probability of error is $prob_{error} \leq 1/2$.

To get $\varepsilon \leq 1/3$ for $MATCH'_n$, we can repeat $PROT'$ twice. Or, we can perform the following protocol: $MATCH'_n$ returns 1 with probability 1/3, and run $PROT'$ with probability 2/3. If the answer is 0, $PROT'$ will always gamble on 0, so the probability of the $MATCH'_n$

returning 1 is 1/3. If the answer is 1, the probability of $MATCH'_n$ returning 1 is $1/3 \cdot 1 + 2/3 \cdot 1/2 = 2/3$. So, in either case, the error rate is $\leq 1/3$.

20.1.1.2 Reduction from $DIST_m$ to $MATCH'_n$

Consider a graph of $3m$ vertices divided into m triplets. Define a numbering on the vertices of each triplet. Alice gets $\{x_1, x_2, \ldots, x_m\}$ and Bob gets $\{y_1, y_2, \ldots, y_m\}$. For the ith triplet, Alice chooses vertex j if $x_i = j$, and Bob chooses edge $\{1,2,3\} \backslash \{k\}$ if $y_i = k$. Run $MATCH'_n$ on this graph.

20.1.1.3 Reducing $DISJ_m$ to $DIST_m$

Reduce $DISJ_m(x, y)$ to $DIST_m(x, y')$ as follows.

$$y'_i = \begin{cases} 2 & \text{if } y_i = 0 \\ 1 & \text{if } y_i = 1 \end{cases}$$

■

Therefore, $D_M(MATCH_n) = \Omega(n)$.

Exercise:

Obtain a lower bound on monotone depth of perfect matching PM_n. (A perfect matching matches all vertices of the graph.)

■

Exercise:

Show that $D_M(MATCH_n) = O(n)$. Also, what is the best constant that you can obtain for $O(n)$?

■