REU 2006 · Discrete Math · Lecture 13

Instructor: László Babai
Scribe: Duru Türkoğlu
Editor: Elizabeth Beazley

July 31, 2006. Last updated August 1, 2006 at 2:30 p.m.
NOT PROOF-READ

13 Communication Complexity

13.1 Testing Equality of Strings

There are two computers on the opposite ends of the Earth. We will name the computers Alice and Bob. Alice and Bob are given information protecting the security of mankind consisting of strings of 0's and 1's, denoted X and Y, respectively. Little green aliens are attacking the Earth by changing the information contained in X and Y. They can change any characters in either of the strings independently. As the green men approach the Earth, our only defense is to verify whether $X = Y$, after the little green men have augmented these strings of data. If we fail in this task, the green aliens will blow up the Earth.

Let us assume that Alice and Bob have unlimited computational power; *i.e.*, they can perform any internal computations for free. Nevertheless, they incur a cost for sending messages to each other, at a price of \$1 per bit. Yao proved that the problem defined above requires at least n bits of communication for n-bit strings X and Y. For instance, if Alice and Bob are given 10^{15}-digit (decimal) numbers, the best algorithm would be that Alice transmits all those 10^{15} digits to Bob, and Bob checks these digits against those of Y, outputting an answer as to whether or not they agree. Directly testing whether or not $X = Y$ thus becomes quite costly.

So that we can save the Earth within our financial means, we are interested in finding a more cost-efficient algorithm. There is a far more efficient randomized protocol to determine whether $X = Y$. The above example requires only 200 digits of communication using this randomized protocol which is given as follows:

1. Alice: Generates a random prime p with 100 decimal digits (initial zeros permitted.)
2. Alice: Calculates $X \mod p$.
3. Alice $\rightarrow$ Bob: Sends p and $X \mod p$.
4. Bob: Compares $X \mod p$ and $Y \mod p$.
   if $X \not\equiv Y \pmod{p}$ then NO
   else YES

The first, second and the fourth steps are free of cost since they are internal computations, so the only cost is the second step which is the transmission of 200 digits. Clearly this algorithm makes errors but the below claim states that the it is very unlikely to make an error:

Claim 13.1.1. $P(\textit{error}) = \begin{cases} 0 & \textit{if } X = Y \\ < \frac{1}{10^{80}} & \textit{if } X \neq Y \end{cases}$

Here, we compute the error using worst case analysis. That is, we assume that the little green aliens know the strategy that we are going to use to test X and Y before they make changes to these strings. The only thing that the aliens do not know is the value of p we will use.

Obviously in the first case there is no error, we simply output the correct answer that they are equal. Thus we shall assume $X \neq Y$. Then the only case of error is: $p \mid X - Y$. Therefore

$$P(\text{error}) = \frac{\nu(X - Y)}{\pi(10^{100})}$$

where $\nu(a)$ is the number of distinct prime factors of a and $\pi(a)$ is the number of primes $\leq a$.

Lemma 13.1.2. *If a is an ℓ-digit integer, then $\nu(a) \leq \ell + 1$.*

Proof. If $\nu(a) = k$ then

$$a \geq \underbrace{2 \cdot 3 \cdot 5 \cdot 7}_{3 \text{ digits}} . \underbrace{11 \cdot 13 \cdots p_k}_{k-4 \text{ digits}}$$

□

So for the claim we have

$$\pi(10^{100}) \approx \frac{10^{100}}{\ln 10^{100}} = \frac{10^{98}}{\ln 10} \approx \frac{10^{98}}{2.3}$$

and by the help of the lemma

$$P(\text{error}) \leq \frac{10^{15}+1}{\frac{10^{98}}{2.3}} = \frac{2.3}{10^{83}} < \frac{1}{10^{80}}$$

This protocol is designed by Rabin, Yao and Simon (Janos Simon of University of Chicago.)

13.1.1 General Results

Yao proved $C(X =^? Y) = n$, where C denotes the communicational complexity with no errors allowed; *i.e.*, deterministic communicational complexity. For the randomized complexity we want to prove something of the following form

$$C_\varepsilon(X =^? Y) = \cdots \log n$$

where ε is the error probability. In other words, we would like to predict the right answer with high probability by just using approximately $\log n$ bits of communication.

Now we can design the detailed version of the protocol while analyzing it. We have the probability of the error defined before

$$P(\text{error}) = \frac{\nu(X-Y)}{\pi(2^s)}$$

where s is the number of bits of the random prime p and X, Y have n binary digits. We have a better bound for the ν function:

Exercise 13.1.3. *$\nu(a) \leq c \cdot \frac{\log a}{\log \log a}$ for some constant c.*

Using this exercise,

$$P(\text{error}) \approx \frac{c \cdot \frac{n}{\log_2 n}}{\frac{2^s}{s \ln 2}} = c \ln 2 \cdot \frac{n}{2^s} \cdot \frac{s}{\log_2 n}$$

We want to make the error less than ε. Choose $s = \log_2 n + r$, so that $\frac{n}{2^s} = \frac{1}{2^r}$ and $\frac{s}{\log_2 n} = 1 + \frac{r}{\log_2 n} < 2$, if we choose $r < \log n$. Then we have

$$P(\text{error}) < c \ln 2 \cdot 2 \cdot \frac{1}{2^r} = \frac{c'}{2^r}$$

for some new constant c'. To have $P(\text{error}) < \varepsilon$, we can set $r := \log\left(\frac{c'}{\varepsilon}\right)$, which makes $r = o(\log n)$ and $s \sim \log_2 n$.

Exercise 13.1.4. *Work out the details for* $\varepsilon = \frac{1}{10^{50}}$.

13.2 Lower Bound Analysis

We begin with some definitions:

Definition 13.2.1. A *boolean function* is a function $f : A \to \{0, 1\}$, where A is any set.

Definition 13.2.2. We say that an input is a *boolean input* when $A = \{0, 1\}^n$.

We can think of a target function f, a boolean function with boolean input: $f : \{0, 1\}^{2n} \to \{0, 1\}$. Now, we assume that A(lice) has access to the first n bits of the input $X = (x_1, x_2, \ldots, x_n) \in \{0, 1\}^n$ and similarly B(ob) has access to the last n bits $Y = (y_1, y_2, \ldots, y_n) \in \{0, 1\}^n$. They are trying to compute f according to the communication protocol described below:

$A \to B$	message string a_1
$B \to A$	message string b_1
$A \to B$	message string a_2
$\vdots$	$\vdots$
A or B	"answer"

Here the a_i's, (resp. b_i's,) are functions of X (resp. Y) and the strings $b_j,\ j \leq i-1$ (resp. $a_j,\ j \leq i-1$). Now for each protocol we can define a function P which computes the "answer" $P(X,Y)$ through communication. P is said to be *correct* if

$$(\forall X, Y) \quad P(X,Y) = f(X,Y) \tag{13.2.1}$$

The *cost* of P is given by # bits communicated (for the worst X, Y.)

Definition 13.2.3. The *communicational complexity* of f, denoted $C(f)$, is defined as follows

$$C(f) = \min_P \text{cost}\,\{P \mid P \text{ correctly computes } f\} \tag{13.2.2}$$

$$= \min_P \max_{X,Y} \{\text{cost of } P(X,Y)\} \tag{13.2.3}$$

Observation 13.2.4. $C(f) \leq n+1$

Theorem 13.2.5 (Mehlhorn–Schmidt). *Define the matrix* $M_f = (f(X,Y))_{2^n \times 2^n}$. *The communicational complexity is bounded from below as follows*

$$C(f) \geq \log_2 rk(M_f) \tag{13.2.4}$$

Example 13.2.6. If $f(X,Y) = \begin{cases} 1 & \text{if } X = Y \\ 0 & \text{else} \end{cases}$, then we have $M_f = Id_{2^n \times 2^n} \Rightarrow$ $\text{rk}(M_f) = 2^n$ which implies $C(f) \geq n$.

Proof of the Mehlhorn–Schmidt Thm. Suppose $C(f) = s$. Therefore there exists a correct protocol P for f which uses s bits of communication. So the # of possible communication strings $z = a_1 b_1 a_2 b_2 \ldots$ is 2^s. Define a function $\underline{P}$ on input (X,Y) to output the whole communication string $z(X,Y)$. Now look at $\underline{P}^{-1}(z) = \{(X,Y) \mid \underline{P} \text{ generates communication } z\}$.

Claim 13.2.7. $\underline{P}^{-1}(z)$ *is a rectangle.*

Lemma 13.2.8. *If* $\underline{P}(X_1,Y_2) = \underline{P}(X_2,Y_1) = z$*; i.e., the whole communication is the same for inputs* (X_1,Y_2) *and* (X_2,Y_1)*, then* $\underline{P}(X_1,Y_1) = \underline{P}(X_2,Y_2) = z$*. That is, the communication is the same for the entire rectangle* $\{X_1,X_2\} \times \{Y_1,Y_2\}$.

	Y_1	Y_2
X_1	$\square = z$	z
X_2	z	$\square = z$

Proof. The proof of the lemma is inductive. Given that $a_1(X_1, Y_2) = a_1(X_2, Y_1)$, we can clearly say that $a_1(X_1, Y_1) = a_1(X_1, Y_2) = a_1(X_2, Y_1) = a_1(X_2, Y_2)$. The first equality holds since A has only access to X, and in this case they are the same. Similarly for the third equality. Inducting on the history, since the history is same for all of them, the next step only depends on the portion of the input they can access. The same trick applies then, and therefore we can conclude that the whole communication is the same for all of them. □

Claim 13.2.7 now follows, since in this case we have homogenous rectangles (rectangles with all entries equal) in the matrix. We now require the following:

Claim 13.2.9. $C(f) = \lceil \log_2(\textit{size of min. partition into hom. rectangles}) \rceil$

Example 13.2.10. For the above matrix, the minimal number of homogeneous rectangles we can use is 5.

We shall prove Claim 13.2.9 through a series of observations. Let $R =$ size of min. partition into hom. rectangles. $C(f) \geq \log_2 R$ because message strings provide a partition into rectangles. Hence there exists a cover with $\leq 2^s$ rectangles.

Exercise 13.2.11. *Prove the other direction upto a factor of 3.*

Now we know $C(f) \geq \log_2 R$ and claim that $R \geq \mathrm{rk}(M_f)$.

Lemma 13.2.12. $rk(A + B) \leq rk(A) + rk(B)$

Proof. Note that span(rows of $A+B$) $\subseteq$ span(rows of A and rows of B). □

Thus Claim 13.2.9 now follows, since $M_f \leq \sum_{\leq R}$ "rectangle matrices" where each "matrix" in the sum has rank 1. This completes the proof of the Mehlhorn–Schmidt theorem. □

A natural question to ask now is the following: Is the bound appearing in the Mehlhorn–Schmidt Theorem tight up to a constant exponent? The below conjecture suggests that the bound is tight, and this problem remains OPEN today.

Conjecture 13.2.13 (log-rank Conjecture (Lovász - Saks)). *There exists a constant c satisfying*

$$(\log_2 rk(M_f))^c \geq C(f) \tag{13.2.5}$$