CS 4510 : Automata and Complexity
Randomized Communication Complexity

Subrahmanyam Kalyanasundaram

April 26, 2010

We shall see a randomized algorithm for equality, which has an $O(\log n)$ communication complexity. We need the following basic theorem from algebra.

Theorem 1. *A univariate polynomial of degree d over $\mathbb{F}$ has at most d roots, unless it is the zero polynomial.*

We saw in class that any deterministic algorithm requires a communication complexity of n to compute the equality function. Here we use a *randomized algorithm*. What this means is that the algorithm would perform differently for the same inputs. You could think of the computer tossing a coin, and deciding between several possible computation paths. Unlike in nondeterminism, here we want a good probability of success, we want that the algorithm succeeds on most inputs.

Alice and Bob have two numbers $a, b \in \{0,1\}^n$ respectively. They need to check if $a = b$. For the algorithm, they decide[1] on a prime number p such that $n^2 \leq p \leq 2n^2$. Alice and Bob may view their numbers in the following manner. Let $a = a_0a_1a_2 \ldots a_{n-1}$ and $b = b_0b_1b_2 \ldots b_{n-1}$ be the bit expansions of a, b. Define polynomials $A(x) = a_0 + a_1x + a_2x^2 + \ldots + a_{n-1}x^{n-1}$ and $B(x) = b_0 + b_1x + \ldots + b_{n-1}x^{n-1}$ over $\mathbb{F}_p$. (Evaluating a polynomial over $\mathbb{F}_p$ can be thought of as evaluating over integers and then taking modulo p.) The protocol is the following:

- Alice picks a random r from $\mathbb{F}_p$, the finite field with p elements.
- Alice computes $A(r)$ and sends $r, A(r)$ to Bob.
- Bob computes $B(r)$ and checks if $A(r) = B(r)$.
- If they are equal, Bob sends 1 back to Alice, else he sends a 0 back.

Notice that if $a = b$, then $A(x) = B(x)$ and hence $A(r) = B(r)$. So the protocol always succeeds, irrespective of the choice of r. However, when $a \neq b$, there is a chance that $A(r)$ and $B(r)$ evaluates to the same number. But we shall show that this is not that likely to happen. That is

$$\text{Prob}[A(r) = B(r) \mid a \neq b] \leq \frac{1}{n}$$

[1]Primes always exist between a number and twice that number.

If $a \neq b$, then the polynomials $A(x) \neq B(x)$, so the polynomial $C(x) = A(x) - B(x) \not\equiv 0$. By theorem 1, $C(x)$ has at most $n-1$ roots. That is, there are at most $n-1$ choices for r which would have made $A(r) = B(r)$. The total number of choices for r is $p = O(n^2)$. So the probability of choosing a *bad* r is at most $1/n$. Hence when $a \neq b$, the protocol succeeds with probability $\geq 1 - 1/n$. This concludes the proof of correctness.

Note that all that needs to be communicated is $r, A(r)$ and a $0/1$ bit. $r, A(r)$ are elements of $\mathbb{F}_p$ and can be communicated using $\log p \leq 2\log n$ bits. So the total communication required is $\leq 4\log n + 1$ which is $O(\log n)$.

Next, we state a generalization of Theorem 1.

Theorem 2. (Schwartz Lemma) *Let $P(x_1, x_2, \ldots, x_n)$ be a nonzero polynomial over n variables with degree at most d over a finite field $\mathbb{F}$. Then for any set $S \subseteq \mathbb{F}$, there are at most $d|S|^{n-1}$ n-tuples $(a_1, a_2, \ldots, a_n) \in S^n$ which satisfy $P(a_1, a_2, \ldots, a_n) = 0$.*

This theorem is stated in Sipser as Lemma 10.15 (page 379). Refer to Sipser for the proof.