REU 2006 · Discrete Math · Lecture 17

Instructor: László Babai
Scribe: Megan Guichard
Editors: Duru Türkoğlu

August 9, 2006. Last updated August 12, 2006 at 7:30pm.
NOT PROOF-READ

17 Boolean Circuits (continued)

17.1 Correction

First, a correction to last time. We said that we could construct Hadamard matrices using prime numbers, which is true, but we used the wrong kind of prime numbers. The correct statement is

Claim 1. If p is a prime number with $p \equiv -1 \pmod 4$, then there exists a $(p+1) \times (p+1)$ Hadamard matrix, given as follows. Let C be the $p \times p$ matrix with (i,j) entry given by $\left(\frac{i-j}{p}\right)$. Form a matrix H by adding an initial row and an initial column of 1s to C, and then replacing 0s in the diagonal by -1s. Then H is Hadamard.

17.2 Boolean circuits (continued)

Theorem 2 (Yao, Håstad). If PARITY is computed by a Boolean circuit of depth d, then size is

$$> 2^{\frac{1}{10} \cdot n^{\frac{1}{d-1}}}.$$

We will approach this by proving a stronger result:

Theorem 3 (Razborov-Smolensky). This lower bound holds even if we permit MOD_3 gates (in addition to AND, OR, NOT gates.)

Corollary 4. The size is $> 2^{c \cdot n^{1/2d}}$.

Definition 5. A MOD_3 gate with inputs $a_1, a_2, \ldots, a_n$ outputs the following:

$$\mathrm{MOD}_3(a_1, a_2, \ldots, a_n) = \begin{cases} 1 & \text{if } \sum a_i \equiv 0 \pmod 3 \\ 0 & \text{o/w} \end{cases} \tag{17.2.1}$$

Definition 6. A MOD_3 circuit is a circuit with boolean gates AND, OR, NOT and MOD_3 gates.

Recall from last time:

Theorem 7. (Razborov-Smolensky) Even using MOD_3 gates, MOD_2 (PARITY) cannot be computed in bounded depth with polynomial size.

Our plan: First, we will show if a Boolean function f can be computed by a MOD_3 circuit of depth d and size m, then f is approximately equal to a polynomial g over $\mathbb{F}_3$ of low degree. This is the key idea: we will switch from circuits, which are elusive combinatorial objects, to algebra, which has much more structure.

Here $f \approx g$ means that

$$\Pr_{x \in \{0,1\}^n}(f(x) \neq g(x)) \text{ is small } \left(< \frac{m}{2^{k^d}} \right)$$

where k is the degree of g. "Low degree" will mean $o(\sqrt{n})$ (little o). Also, there is a tradeoff between the approximation and the degree, the higher the degree, the less the error in approximation. For future analysis on the degree, note $k^d \approx 2^{\sqrt{n}}$ means $k \approx 2^{n^{1/2d}}$ (n is the number of variables.)

Second, we will show that if PARITY is approximately equal to a low-degree polynomial over $\mathbb{F}_3$, then *every* boolean function f is approximately equal to a polynomial of degree $\frac{n}{2} +$ little.

Finally, we will show by counting that this is impossible, leading to the desired contradiction.

Proof. <u>First Part:</u> Let a_i be boolean variables (variables taking values 0 or 1.) Then, in any field, $a_1 \wedge a_2 \wedge \cdots \wedge a_k = a_1 \cdot a_2 \cdots a_k$. Recall also deMorgan's rule:

$$a_1 \wedge \cdots \wedge a_k = \overline{\overline{a_1} \vee \cdots \vee \overline{a_k}}$$

and similarly

$$b_1 \vee \cdots \vee b_k = \overline{\overline{b_1} \wedge \cdots \wedge \overline{b_k}}$$

Also, $\overline{a} = 1 - a$.

Therefore we see that we can easily switch between AND and OR gates. Now, if we naively substitute all the AND and OR gates as we formulate above (the product,) the

degree gets huge. If you imagine the AND gate with inputs $a_1, a_2, \ldots, a_k$ (the quantity of the inputs is called **fan-in**), as the fan-in gets larger we get larger exponent in our polynomial for this gate. We then need some kind of reduction in the degree, for this we will compute the result approximately.

Since handling AND and OR gates are the same (conversion is linear) we shall consider $b_1 \vee \cdots \vee b_k$. The quantity $\left(\sum_{i \in I} b_i\right)^2 \pmod 3$ might be a good approximation, where I is a random subset of $\{1, \ldots, k\}$. So we are investigating whether

$$b_1 \vee \cdots \vee b_k \approx \left(\sum_{i \in I} b_i\right)^2 \pmod 3$$

Some observations:

1. The right-hand side is boolean (equal to 0 or 1), because all squares are 0 or 1 mod 3.
2. Consider the left-hand side. It is 0 only if all b_i are zero, in which case the right-hand side is also 0, so $\Pr_I(\text{error}) = 0$ Here the probability of error is given by worst-case analysis and randomization is over the subset I (one can think that there is an adversary which provides the worst inputs $b_1, b_2, \ldots, b_k$ and we hope that our random subset will help us providing the correct result.)
3. If the left-hand side is 1, then $\Pr_I(\text{right-hand side} = 0) \leq \frac{1}{2}$. This can be seen as follows. Suppose that $b_\ell = 1$. Pair up the subsets of $\{1, \ldots, k\}$ by pairing a subset with the subset given by toggling membership of ℓ in the set. Then, for every pair I and I', at least one of I, I' gives the correct answer on the right-hand side. So, for every input $b_1, \ldots, b_k$,

$$\Pr_I(b_1 \vee \cdots \vee b_k \neq \left(\sum b_i\right)^2 \pmod 3) \leq \frac{1}{2}$$

as desired.

If we do this multiple times, say N times, using random sets I_j each time, then it's a good bet that

$$b_1 \vee \cdots \vee b_k = \bigwedge_{j=1}^{N} \left(\sum_{i \in I_j} b_i\right)^2.$$

That is, the probability of error is $\leq \frac{1}{2^N}$.

This means that we can replace all AND and OR gates by polynomials of degree $2k$, because the expression on the right-hand side is a polynomial with this degree; we are replacing each variable by a polynomial of degree 2.

So, we have a new randomized circuit such that

$$(\forall x) \Pr(f(x) \neq g(x)) \leq \frac{m}{2^k} \tag{17.2.2}$$

where $g(x)$ refers to the new circuit, m is the size of the original circuit, and the inequality is given by the union bound. The polynomial g was chosen at random; we can repeat this for each gate in the original circuit. Since the original circuit had depth d, g will be a polynomial of degree $\leq (2k)^d$.

Now, since Eq. 17.2.2 holds $\forall x$, it follows that

$$\Pr_{x,g}(f(x) \neq g(x)) \leq \frac{m}{2^k} \tag{17.2.3}$$

and so

$$(\exists g) \Pr_x(f(x) \neq g(x)) \leq \frac{m}{2^k} \tag{17.2.4}$$

Indeed, we use the traditional trick which we have applied also in randomized vs. distributional communication complexity. Now, we have moved to a fixed polynomial which approximates f in the sense that not every input is correctly computed, but the fraction which are incorrectly computed is at most $m/2^k$.

In this sense, $f(x) \approx g(x)$; g is low-degree, meaning $(2k)^d$. f was a function computable in depth d and size m. The $\approx$ means that $\Pr_x(\text{error}) \leq \frac{m}{2^k}$.

This finishes the first part of the plan.

Recall that $\text{PARITY}(x_1, \ldots, x_n) = \sum x_i \pmod 2$. Suppose that PARITY $\approx g(x_1, \ldots, x_n)$ over $\mathbb{F}_3$. Let

$$y_i = 1 - 2x_i = \begin{cases} -1 & \text{if } x_i = 1 \\ 1 & \text{if } x_i = 0 \end{cases}$$

Then $x_i = -\frac{1}{2}(y_i - 1)$, so $g(x_1, \ldots, x_n) = h(y_1, \ldots, y_n)$, where g and h have the same degree. (In $\mathbb{F}_3$, $-\frac{1}{2} = 1$ therefore $x_i = y_i - 1$.)

Now $\prod_{i=1}^n y_i = (-1)^{\text{PARITY}(\underline{x})}$. Notice that $(-1)^z = 1 - 2z$, so

$$\prod_{i=1}^n y_i = 1 - 2 \cdot \text{PARITY}(\underline{x}) \approx 1 - 2g(\underline{x}) = 1 - 2h(\underline{y}),$$

and this is a low-degree polynomial. Now let $f : \{0,1\}^n \to \{0,1\}$ be any boolean function.

Lemma 8. f can be written as a **multilinear** polynomial over any field F. Multilinear means each variable appears with monomial at most 1, i.e. $cx_1x_3x_4$ is multilinear but $cx_1^2x_3x_4$ is not.

Proof. Base Case: Given $\underline{a} \in \{0,1\}^n$, define

$$f_{\underline{a}}(\underline{y}) = \begin{cases} 1 & \text{if } \underline{y} = \underline{a} \\ 0 & \text{otherwise} \end{cases}$$

Then $f_{\underline{a}}(\underline{y}) = \prod t_i$, where

$$t_i = \begin{cases} y_i & \text{if } a_i = 1 \\ -y_i & \text{if } a_i = 0 \end{cases}$$

Thus f_a is multilinear. Now every function $\{0,1\}^n \to F$ is a linear combination of the f_a, and the lemma follows. □

Returning to the main proof, if f is *any* Boolean function, then f is a multilinear polynomial over $\mathbb{F}_3$. Write

$$f = \sum_{I \subset \{1,\ldots,n\}} \alpha_I \prod_{i \in I} x_i = \sum_I \beta_I \prod_{i \in I} y_i.$$

Now, for $|I| \leq \frac{n}{2}$, do nothing. For $|I| > \frac{n}{2}$, write

$$\prod_{i=1}^{n} y_i = \prod_{i \in I} y_i \prod_{i \notin I} y_i.$$

Thus

$$\prod i \in I y_i = \prod_{i \notin I} y_i \prod_{i=1}^{n} y_i$$

where the first term is of degree $\leq n/2$ and the second term is approximately equal to a polynomial of low degree. There is only one approximation, and we can see that the probability of error in computing $\prod_{i \in I} y_i$ is still small. Also, once we translate back to the original variables x_i, every x_i is either 0 or 1, so $x_i^2 = x_i$, so we may assume the polynomial is multilinear. (Alternately, every y_i squares to 1 and a similar result holds.) This finishes part 2 of the plan; every Boolean function is approximable by a multilinear polynomial of degree $\leq \frac{n}{2} +$ something little.

Finally, we will do some counting. The space of polynomials of degree $\leq \frac{n}{2} + \frac{\sqrt{n}}{100}$ has dimension

$$\sum_{j=0}^{n/2+\sqrt{n}/100} \binom{n}{j} = 2^{n-1} + \frac{2^n}{large} = (1+\epsilon)2^{n-1}.$$

What is the number of Boolean functions "close" ($\epsilon 2^n$-close) to a given Boolean function? That is, we want the size of the set

$$\{g : \Pr_x(f(x) \neq g(x)) \leq \epsilon\}$$

which is

$$\sum_{j=0}^{\epsilon 2^n} \binom{2^n}{j}.$$

Exercise 9. Show that $\sum_{j=0}^{k} \binom{n}{j} < \left(\frac{en}{k}\right)^k$.

Using the exercise, we see

$$\sum_{j=0}^{\epsilon 2^n} \binom{2^n}{j} < \left(\frac{e2^n}{\epsilon 2^n}\right)^{\epsilon 2^n} = \left(\frac{e}{\epsilon}\right)^{\epsilon 2^n}$$

and for all ϵ there exists δ such that this last quantity is $< (1+\delta)^{2^n}$.

Therefore, the total number of approximable functions is less than

$$3^{(1+\epsilon)2^{n-1}} \cdot (1+\delta)^{2^n} < 3^{(1+\epsilon')2^{n-1}} << 2^{2^n}$$

(using the approximation $\log_2 3 < 2$).

This completes the proof. □