University of California, Los Angeles
CS 281 Computability and Complexity

Instructor: Alexander Sherstov
Scribe: Oliver Broadrick
Date: January 10, 2023

LECTURE

1

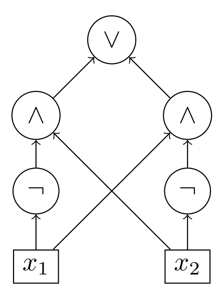
An Invitation

In the first two lectures we motivate, state, and prove the classic Razborov-Smolensky Theorem, illustrating the beauty of Computational Complexity and using some techniques common throughout this course. The theorem, a "gem" of the area, says that there is no constant depth and polynomial size circuit computing the parity function, $\bigoplus_{i=1}^{n} x_i$.

1.1 Basic Notations

A (boolean) circuit is a directed acyclic graph. Nodes of in-degree zero are *input* nodes and are elements of $\{x_1, \ldots, x_n\}$. There is one node of out-degree zero, the *output* node. All other nodes are *gates* and are labeled by elements of $\{\wedge, \vee, \neg\}$ ($\neg$-gates have in-degree one). The size of a circuit is the number of nodes in it. The depth of a circuit is the length of the longest path from input to output. A circuit C computes a function in a natural way, defined inductively as follows. Input nodes x_i compute x_i, $\vee$-gates (resp. $\wedge$-gates) with children $C_1, \ldots, C_k$ compute $\bigvee_i C_i$ (resp. $\bigwedge_i C_i$), and $\neg$-gates with child C compute $\neg C$.

EXAMPLE 1.1. The following circuit computes $x_1 \oplus x_2$.



This circuit has depth 3 and size 7.

A computational problem is a language $L \subseteq \{0,1\}^*$.

Definition 1.2. A circuit family $\{C_n\}_{n=1}^{\infty} = (C_1, C_2, \ldots, C_n, \ldots)$ computes language $L \subseteq \{0,1\}^*$ if, for all $n \in \mathbb{N}$ and all $x \in \{0,1\}^n$,

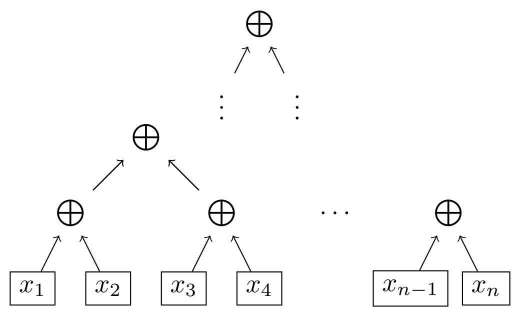
$$C_n(x) = \begin{cases} 1 & \text{if } x \in L \\ 0 & \text{if } x \notin L. \end{cases}$$

1.2 Parity

An important language is the parity language which contains all binary strings of odd hamming weight. That is,

$$L_{\text{PARITY}} = \{x \in \{0,1\}^* : x \text{ has odd parity}\} = \{1, 10, 01, 111, 100, 010, 001, \ldots\}.$$

If we let $\oplus$ denote the circuit from Example 1.1, then the following circuit computes L_{PARITY}.



Note that this circuit has depth $O(\log n)$ and size $O(n)$. Could its depth be reduced further while maintaining its poly(n) size? We now define AC^0, the class of polynomial-size circuits with *constant* depth.

Definition 1.3. $L \in AC^0$ if L is computable by a circuit family $(C_1, C_2, \ldots, C_n, \ldots)$ for which there exists a constant $c > 0$ such that for all $n \in \mathbb{N}$

- size$(C_n) \leq n^c$,
- depth$(C_n) \leq c$, and
- C_n has arbitrary fan-in.

The fan-in of a circuit is the maximum in-degree of its gates. We allow arbitrary fan-in because if the fan-in were bounded, e.g. by 2, then only constantly many (2^c) input variables could possibly be mentioned in the circuit.

1.3 Algebraization

The Razborov-Smolensky Theorem applies to even stronger circuits, those with an additional type of gate. Define the MOD$_m$ gate for positive integer m as

$$\text{MOD}_m(x_1, \ldots, x_k) = \begin{cases} 1 & \text{if } \sum_i x_i \equiv 0 \mod m \\ 0 & \text{else.} \end{cases}$$

We have now defined the necessary objects to begin working towards the main theorem. Our proof will consist of two lemmas. The first will show that $\{\wedge, \vee, \neg, \mathrm{MOD}_m\}$-circuits (for m prime) can be well approximated by low degree polynomials. In particular, the approximation will be correct on a proportion of the possible inputs which grows exponentially as the degree of the polynomial grows polynomially. On the other hand, the second lemma will show that the parity functions form a basis for the space of such functions and so cannot be accurately captured by such low-degree approximations.

Lemma 1.4 (Razborov-Smolensky). *Let m be a prime, C be a $\{\wedge, \vee, \neg, MOD_m\}$-circuit, and $t \in \mathbb{Z}^+$ a parameter to be tuned. Then there exists a polynomial $p \in \mathbb{F}_m[x_1, \ldots, x_n]$[1] such that*

- $\mathbb{P}_x[p(x) = C(x)] \geq 1 - \dfrac{size(C)}{m^t}$, *and*
- $deg\ p \leq (tm)^{depth(C)}$.

Note that for $O(\log n)$-depth circuits, this lemma promises a degree $O(n)$ polynomial, which exists trivially (with zero approximating error), and so this lemma is specifically helpful with constant depth circuits.

Proof. We construct the approximating polynomial $p(x)$ inductively from the bottom up, starting with the leaves. First preprocess the circuit by converting $\wedge$ nodes to $\vee$ and $\neg$ nodes using De Morgan's Law, (i.e. convert every $C \wedge C'$ to $\neg(\neg C \vee \neg C')$).

- For an input node x_i, let $p(x) = x_i$, incurring no error and with degree 1.
- For a $\neg$-gate whose input is approximated already by $q(x)$, let $p(x) = 1 - q(x)$, incurring no error and no change in degree.
- For a MOD_m-gate with inputs approximated by $q_1(x), q_2(x), \ldots, q_k(x)$, we compute $p(x) = 1 - (\sum_i q_i(x))^{m-1}$. If $\sum_i q_i(x) = 0 \mod m$, then $p(x) = 1$. If $\sum_i q_i(x) \neq 0 \mod m$, then by Fermat's Little Theorem, $p(x) = 1 - 1 = 0$.[2] Again, we incur no error, and we increase the degree multiplicatively by m (among friends).
- For an $\vee$-gate with inputs approximated by $q_1(x), q_2(x), \ldots, q_k(x)$, we choose $a_1, \ldots, a_k \in \mathbb{F}_m$ uniformly at random and observe

$$\mathbb{P}_{a_1,\ldots,a_k}[a_1 q_1 + \ldots + a_k q_k = 0] = \begin{cases} 1 & \text{if } \bigvee_i C_i(x) = 0 \\ 1/m & \text{if } \bigvee_i C_i(x) = 1. \end{cases} \tag{1.1}$$

  Clearly if all q_i's are zero, then the sum is zero. If some q_i is nonzero (i.e. one), then the overall sum is only nonzero if $a_i \equiv -\sum_{j \neq i} a_j q_j \mod m$, and so with probability $1/m$. We again use Fermat's Little Theorem to cast nonzero outcomes to one. Additionally, we use the parameter t to trade off between accuracy and degree by computing t such random weighted sums. With uniformly random $a_{ij}\mathbb{F}_m$ as the ith coefficient of the jth such sum, let

$$p(x) = 1 - \prod_{j=1}^{t} \left(1 - \sum_i (a_{ij} q_i)^{m-1}\right),$$

[1] $\mathbb{F}_m = GF(m)$.

[2] $\sum_i q_i(x)$ could be any nonzero element in $\mathbb{F}_m$, and so we are using Fermat's Little Theorem (i.e. raising this sum to $m-1$) in order to "cast" all nonzero elements back to 1.

with a multiplicative increase in degree of tm (among friends) and probability of error (over the choice of a_{ij}'s) at most m^{-t}:

$$\mathop{\mathbb{P}}_{a_{11},\ldots,a_{kt}}\left[p(x) \neq \bigwedge_i C_i(x)\right] \leq m^{-t}.$$

(This point marks the end of lecture one and the start of lecture two.)

Let P be the polynomial for the whole circuit C defined inductively by these rules. Note that P is a random variable (the a_{ij}'s at each $\vee$-gate are selected at random). What are the degree and accuracy of P as an approximating polynomial for C? First, every inductive rule increases the degree of the approximating polynomial by at most tm, and so

$$\deg(P) \leq (tm)^{\mathrm{depth}(C)}.$$

Now we focus on accuracy. Fix some $x \in \{0,1\}^n$ and consider some gate g in the circuit computing $g(x)$ and with approximating polynomial p. Let the children of g be $g_1, \ldots, g_k$, with approximating polynomials $p_1, \ldots, p_k$. Then the probability of an error by p despite faithful inputs is at most m^{-t}:

$$\mathop{\mathbb{P}}_{P}\left[p(x) \neq g(x) \mid \bigwedge_i \mathbb{1}\left[p_i(x) = g_i(x)\right]\right] \leq \frac{1}{m^t}.$$

An overall approximation error ($P(x) \neq C(x)$) implies that there was some gate whose approximating polynomial erred despite faithful inputs. Therefore we conservatively apply the union bound to obtain

$$\mathop{\mathbb{P}}_{P}[P(x) \neq C(x)] \leq \frac{\mathrm{size}(C)}{m^t}.$$

In particular,

$$\mathop{\mathbb{P}}_{P}[P(x) = C(x)] \geq 1 - \frac{\mathrm{size}(C)}{m^t}.$$

However, we are interested in the accuracy as the proportion of inputs $x \in \{0,1\}^n$ for which $P(x) = C(x)$. So, we introduce $x \in \{0,1\}^n$ as a (uniform) random variable and observe that

$$\mathop{\mathbb{E}}_{P}\left[\mathop{\mathbb{P}}_{x}\left[P(x) = C(x)\right]\right] = \mathop{\mathbb{E}}_{x}\left[\mathop{\mathbb{P}}_{P}\left[P(x) = C(x)\right]\right] \geq 1 - \frac{\mathrm{size}(C)}{m^t}$$

where the equality follows because x and P are independent random processes. Since the expected accuracy (over P) satisfies the desired inequality, there exists some particular approximating polynomial satisfying it, as needed. ∎

We now move onto the second lemma which does not involve circuits. Rather, it is a general statement about polynomial approximations of the parity function, relating the degree of such a polynomial to its accuracy.

LEMMA 1.5 (Razborov-Smolensky). *Let m be an odd prime and $p \in \mathbb{F}_m[x_1, \ldots, x_n]$. Let A be the set of points on which p computes the parity function:*

$$A := \left\{x \in \{0,1\}^n : p(x) = \bigoplus_{i=1}^n x_i\right\}.$$

Then,

$$|A| \leq \sum_{i=0}^{\frac{n+deg(p)}{2}} \binom{n}{i}.$$

At a high level, this lemma should seem reasonable. For $\deg(p) = 0$, it promises $|A| \leq 2^{n-1}$, which is half of the 2^n total points and is achievable by either constant function (i.e. $p(x) = 0$ or $p(x) = 1$). On the other hand, if $\deg(p) = n$, then we are promised $|A| \leq 2^n$, the whole hypercube, which is trivially tight because all functions on the boolean hypercube can be written as degree n polynomials by interpolation.

Before proceeding to the proof of the theorem, we give some additional intuition on the density of the boolean hypercube around hamming weight $n/2$. View the boolean hypercube ($\{0,1\}^n$), as in Figure 1.3, where height corresponds to hamming weight and width corresponds to density. Points are highly concentrated at hamming weight $n/2$, with virtually all points have hamming weight within $\sqrt{n}$ of $n/2$. In particular, the proportion of points of hamming weight within $k\sqrt{n}$ of $n/2$ is

$$\frac{\binom{n}{n/2-k\sqrt{n}} + \ldots + \binom{n}{n/2+k\sqrt{n}}}{2^n} = 1 - e^{-\Theta(k^2)},$$

using Sterling's approximation.[3] Similarly, as the degree of an approximating polynomial grows, additional "rows" of the hypercube in this view can be captured, consistent with the bound in the lemma.

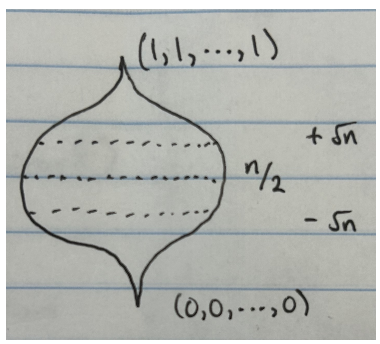


FIGURE 1.1: A common diagram of the boolean hypercube ($\{0,1\}^n$), emphasizing the distribution of hamming weight. Height corresponds to hamming weight and width corresponds to relative density. Nearly all points have hamming weight within $\pm O(\sqrt{n})$ of $n/2$.

We now prove Lemma 1.4.

Proof. Fix any $S \subseteq \{1, \ldots, n\}$. Then, on A,

- $(-1)^{\sum_{i \in S} x_i} = \prod_{i \in S}(1 - 2x_i)$, and
- $(-1)^{\sum_{i \in S} x_i} = \prod_{i=1}^{n}(-1)^{x_i} \cdot \prod_{i \in \bar{S}}(-1)^{x_i} = (1 - 2p(x)) \cdot \prod_{i \in \bar{S}}(1 - 2x_i)$,

[3]See the "Handout on binomial coefficients" on BruinLearn.

where $\bar{S}$ is the complement of S in $\{1, \ldots, n\}$. Note that the first representation has degree $|S|$, where the second has degree $\deg(p) + n - |S|$. Therefore, every function $(-1)^{\sum_{i \in S} x_i}$ is representable as a polynomial of degree

$$d \leq \frac{n + \deg(p)}{2},$$

the mean of the two possible representations' degrees.

However, these (2^n) functions form a basis of the vector space of functions $\mathbb{F}_m^A$. In particular, for any $f : A \to \mathbb{F}_m$, we have

$$\begin{aligned} f(x) &= \sum_{a \in A} f(a) \prod_{i=1}^{n} \frac{1 + (-1)^{x_i + a_i}}{2} \\ &= \sum_{a \in A} f(a) 2^{-n} \prod_{i=1}^{n} (1 + (-1)^{x_i} (-1)^{a_i}) \\ &= \sum_{a \in A} f(a) 2^{-n} \sum_{S \subseteq \{1, \ldots, n\}} (-1)^{\sum_{i \in S} a_i} (-1)^{\sum_{i \in S} x_i}, \end{aligned}$$

where the first equality follows by simply interpolating f on A (the product computes $\mathbb{1}[x = a]$), and the other equalities are simply expanding the product to a sum to make clear that we have found a linear combination as needed. In particular, note that $f(a)2^{-n}$ and $(-1)^{\sum_{i \in S} a_i}$ are scalars w.r.t. x, and so we have written f as a linear combination of functions of the form $(-1)^{\sum_{i \in S} x_i}$ for $S \subseteq [n]$. Since we know that such functions are representable by polynomials of degree $d \leq \frac{n + \deg(p)}{2}$, we have a polynomial computing f on A of degree $d \leq \frac{n + \deg(p)}{2}$. Moreover, we can assume the polynomial is multilinear since $x_i = x_i^2 = x_i^3 = \ldots$ for $x_i \in \{0, 1\}$.

Finally, the number of functions in $\mathbb{F}_m^A$ must be at most the number of multilinear polynomials of degree $d \leq \frac{n + \deg(p)}{2}$. Thus, since the number of monomials in a multilinear polynomial of degree d is exactly the number of ways to choose a subset of the n variables of size at most d, we have

$$m^{|A|} \leq m^{\sum_{i=0}^{\frac{n + \deg(p)}{2}} \binom{n}{i}}.$$

∎

With both lemmas proven, we can now formally state and prove the main theorem.

Theorem 1.6 (Razborov-Smolensky). *Let C be a $\{\wedge, \vee, \neg\}$-circuit that computes*

$$C(x_1, \ldots, x_n) = \bigoplus_{i=1}^{n} x_i.$$

Then,

$$\mathit{size}(C) \geq \frac{1}{4} (1.2009...)^{n^{\frac{1}{2\mathit{depth}(C)}}}.$$

Note that this gives an exponential lower bound on the size of the circuit when the depth is constant, but for $\log n$ depth and greater, it gives no useful bound.

Proof. Let m be an odd prime, $t \in \mathbb{N}$ a parameter. We have

$$1 - \frac{\text{size}(C)}{m^t} \leq \max_{p \in \mathbb{F}_m[x_1,\ldots,x_n]} \left\{ \mathop{\mathbb{P}}_{x \in \{0,1\}^n} \left[p(x) = \bigoplus_{i=1}^{n} x_i \right] \right\} \leq 2^{-n} \sum_{i=0}^{\frac{n+\deg(p)}{2}} \binom{n}{i}$$

where the first and second inequalities follow respectively from Lemmas 1.4 and 1.5. So, removing the middle expression, and substituting $\deg(p) \leq (tm)^{\text{depth}(C)}$, we have

$$1 - \frac{\text{size}(C)}{m^t} \leq 2^{-n} \sum_{i=0}^{\frac{1}{2}(n+(tm)^{\text{depth}(C)})} \binom{n}{i},$$

and so, separating off the first $n/2+1$ terms in the sum as $\frac{1}{2}$, and using Sterling's Approximation to upper bound the remaining terms, we obtain

$$1 - \frac{\text{size}(C)}{m^t} \leq \frac{1}{2} + \frac{(tm)^{\text{depth}(C)}}{2\sqrt{n}},$$

and so

$$\text{size}(C) \geq \left(\frac{1}{2} - \frac{(tm)^{\text{depth}(C)}}{2\sqrt{n}} \right) m^t,$$

from which the result is obtained by setting $m = 3$ and $t = \frac{1}{6} n^{\frac{1}{2\text{depth}(C)}}$. ∎

We close the second lecture with three challenge problems.

1. Which part of the Razborov–Smolensky technique requires that the modulus m be a prime number?

2. Construct a polynomial-size circuit of depth $O(\log n)$ that computes the majority function on n bits.

3. Prove that there is no polynomial-size circuit of constant depth that computes the majority function on n bits.

In the next lecture, we begin "course proper".