1 Introduction

1.1 Administrative Notes

• First HW Questions are posted.
• Lecture 1 scribe notes are posted.

1.2 Last Time

• Shrinkage
• Andreev’s bound [And87]
• Depth reduction via Spira
• Monotone Formulas. Specifically Valiant’s ’84 construction for $O(n^{5.3})$ size $maj$ monotone formula.

1.3 Today

• Finish Valiant ’84 [Val84]
• Communication Complexity (CC) and Formula Complexity
  – CC for functions
  – CC for relations
  – Boolean formulas have direct correspondence with certain communication protocols for certain relations
• Khrapchenko’s lower bound: $\text{size}($PAR$) = \Omega(n^2)$ [Khr71]
Figure 1: a) a single layer of AND and OR nodes, there are $2.65 \log n$ many layers of these. b) the total depth of the tree is $5.3 \log n$. c) The number of ‘slots’ at the bottom level is $n^{5.3}$.

1.4 Next Up

- Lower bound of Karchmer/Wigderson
- $\text{mdepth}($CON$) = \Omega(\log^2 n)$

2 Valiant’s ’84 Majority Formula Construction

Recall from last time, we constructed a tree multiple layers, each layer composed of an ‘AND’ level and an ‘OR’ level. (Figure 1). The entire tree was composed of $2.65 \log n$ layers, bringing the depth of the tree to $5.3 \log n$. Because it is a full binary tree, we have $n^{5.3}$ many leaf nodes, each one with the value:

- 0 with probability $1 - 2\alpha$
- A uniform choice from $x_1, \ldots, x_n$ with probability $2\alpha$

We started to prove that with probability $\geq \frac{1}{2}$, a random formula $F$ with this construction computes $\text{maj}(x)$ correctly for all $x$. If we fix any $x$ and show that

$$\Pr_F[F(x) = \text{maj}(x)] \geq 1 - \frac{1}{2^{n+1}}$$

then using a union bound over all $x$ (note there are $2^n$ many of them)

$$\Pr_F[F(x) \neq \text{maj}(x) \text{ for any } x] \leq \frac{1}{2^{n+1}} \cdot 2^n = \frac{1}{2}$$

which therefore implies that

$$\Pr_F[F(x) = \text{maj}(x) \forall x] \geq \frac{1}{2}$$
To start the analysis we look at one layer at the bottom of the tree (a single AND connecting two OR’s) (Figure 2). We saw last time that if a leaf had probability \( p \) of being labeled ‘1’, the overall probability of the layer being ‘1’ is

\[
A(p) = p^4 - 4p^3 + 4p^2
\]

We use the notation \( A^{(t)}(p) \) to indicate repeated, nested applications of \( A \) to \( p \) (i.e. \( A^{(2)}(p) = A(A(p)) \)). At the top our probability of ‘1’ is \( A^{(2.65 \log n)}(p_x) \).

Fix an \( x \). Because we assigned ‘1’ based in part by randomly drawing a variable from \( x \), this induces a \( p \) for this \( x \): \( p = p_x = \Pr[\text{slot } j \text{ is 1}] \).

\[
p_x = 2\alpha \cdot \frac{\# \text{ of 1’s in } x}{n}
\]

Assume \( n \) is odd,

- if \( x \) is s.t. \( \text{maj}(x) = 1, p_x \geq \alpha + \frac{1}{4n} \)
- if \( x \) is s.t. \( \text{maj}(x) = 0, p_x \leq \alpha - \frac{1}{4n} \)

Say \( x \) is s.t. \( \text{maj}(x) = 1, \) so \( p_x \geq \alpha + \frac{1}{4n} \). With this fixed \( x \), let \( p_0 = p_x \) and \( p_t \) be \( A^{(t)}(p_0) \). We need to show that at the top of the tree, the probability of ‘1’ is large

\[
p_{2.65 \log n + 31} \geq 1 - \frac{1}{2^{n+1}} \quad (1)
\]

There are three stages (a useful figure to keep in mind is Figure 3):

1. “rapid divergence from \( \alpha \) when \( p_t \) close to \( \alpha \)”
   - let \( p_t = \alpha + \epsilon_t, \epsilon_0 \geq \frac{1}{4n} \)
   - (HW 2.a) For \( t = 1.625 \log n \) have \( p_t \geq \alpha + 10^{-6} \)
     - (HW hint) think about the derivative
2. “constant improvement” If \( p_t \geq \alpha + 10^{-6} \), then \( p_{t+31} \geq 1 - \frac{1}{8} \)
3. “rapid convergence close to 1”
   
   • (HW 2.b) If $p_t = 1 - c, c < \frac{1}{8}$, then $p_{t + \log n} \geq 1 - \frac{1}{2^{n+1}}$

After these stages we have that

$$p_{2.625 \log n + 31} \geq 1 - \frac{1}{2^{n+1}}$$

and since

$$2.625 \log n + 31 < 2.65 \log n + 31$$

we know that equation (1) is true.

Note that Valiant’s proof here is non-explicit. AKS had explicit proofs with bounds around $n^{10^{73}}$, which were improved to $n^{6000}$

### 3 Communication Complexity

Before we start to explore the close relationship between the communication complexity (cc) of relations and formula complexity, we will

- define communication complexity and examine the cc of functions via the complexity of protocols that solve those functions.
- define and examine the cc of relations and look at Khrapchenko’s $\Omega(n^2)$ lower bound for the PAR and maj relations, which directly corresponds to a lower bound on formulas for those functions.

Also coming up is Karchmer/Wigderson l.b. of $m\text{depth}(\text{CON}) = \Omega(\log^2 n)$ (note that we don’t get to it in this lecture).

![Figure 3: Plot of $A(p)$. Notice the fixed point of $\alpha$ where $A(p)$ and $y = p$ intersect. $\alpha \approx 0.38$](image-url)
3.1 Communication Complexity of Functions

Consider how much communication two parties with infinite computing power need to compute a joint function of their inputs. This amount of communication is a measure of the complexity of the function.

Let $A$ and $B$ be two cooperative parties, say Alice and Bob. There is a known $f : X \times Y \to Z$, so $f(x, y) = z$. However, $A$ only gets $x$ and $B$ only gets $y$. They must send information to each other to compute $f$ and at the end both must know the result of $f(x, y)$.

**Example 1** (Trivial u.b.). $X, Y = \{0, 1\}^n$, $Z = \{0, 1\}$. We can always compute any $f$ with $n + 1$ bits of communication:

1. $A$ sends all of $x$ to $B$ ($n$ bits)
2. $B$ computes $f$ and sends the output back to $A$ (1 bit)

**Example 2** (Median search). $X, Y = \{0, 1\}^n$, $Z = [n]$. $f(x, y)$ is the median of the multiset $x \cup y$ where we consider $x$ and $y$ as subsets of $[n]$ corresponding to the indices of the ‘1’s in the input bit strings.

$A$ and $B$ can do a binary search for the result. They each maintain an interval $[i, k]$, initially set to $[1, n]$. Each stage consists of:

1. Calculating the midpoint $j = \frac{i + k}{2}$ (no communication required).
2. $A$ sends (# of elements in $x$ that are $\leq j$), (# of elements in $x$ that are $> j$)
3. $B$ uses this to figure out if median $\in [i, j]$ or median $\in [j + 1, k]$. $B$ sends either ‘0’ or ‘1’ to indicate which scenario is true.

They repeat this process until $i = k$ and that value is the median.

$A$’s transmission is $2 \log n$ bits, and $B$’s transmission is 1 bit. Since they cut the possible space of the median in half each time, there are $\log n$ stages. This brings the total communication to $(2 \log n + 1) \cdot \log n = O(\log^2 n)$.

**Example 3** (Equality). $X, Y = \{0, 1\}^n$, $Z = \{0, 1\}$, $f(x, y) = EQ(x, y)$ (i.e. 1 if $x = y$, 0 otherwise)

We’ll see that this has a $n + 1$ lower bound.

**Definition 4.** A protocol is the complete “set of rules” for interaction. $A$ and $B$ both follow the protocol and at the end of executing the protocol, both $A$ and $B$ know the result of $f(x, y)$. 
Definition 5. A communication protocol for $f : X \times Y \rightarrow Z$ is a rooted binary tree where:

- Each internal node $v$ has 2 children, one corresponding to ‘0’ and one to ‘1’
- Each internal node is labeled with either
  - $(A, f_v : X \rightarrow \{0, 1\})$
  - $(B, f_v : Y \rightarrow \{0, 1\})$
- Each leaf is labeled with some $z \in Z$
- Protocol computes $f$ by “walking down the tree together”, at each node each party says the bit he or she computed there.

$\forall (x, y) \in X \times Y$, the label $z$ of the leaf they reach on input $(x, y)$ is $f(x, y)$.

It’s important to note that the functions $f_v$ at each node, do not require the information from the other party as inputs. The information from the other party determines where in the tree we are, and hence which $f_v$ we are executing. Also note that because $A$ and $B$ walk down the tree together, there is no need to send a ‘final answer’ from one to the other. For instance, in the median example above (2), they both realize at the same time when they’ve completed a stage that results in $i = k$, and at that point they both know the median. There is no need to spend an additional $\log n$ bits of communication for $B$ to tell $A$ what the median is.

Definition 6. For a protocol $P$, $\text{cost}(P) = \text{the depth of the protocol tree}$ (this is the number of bits of communication needed in the worst case).

The deterministic communication complexity of $f$, $D(f)$ is the minimum $\text{cost}(P)$, $\forall P$ that compute $f$.

We can view the function $f(x, y)$ as a matrix with the rows corresponding to the space of $X$, the columns to $Y$, and the cells to the value of $f(x, y)$ for the $x$ and $y$ for that row/column (Figure 4). The ordering of the rows and columns isn’t important. By considering the matrix, we have a technique for establishing the lower bound of protocol’s cost by partitioning the matrix into monochromatic rectangles.

Definition 7. A combinatorial rectangle $R$ is a subset $R \subseteq X \times Y$ s.t. for $U \subseteq X$ and $V \subseteq Y$, $R = U \times V$.  


Fix a protocol $P$ and let $v$ be a node in $P$. Let $R_v$ denote the set of $(x, y)$ pairs that reach $v$. Note that $\{R_\ell\}_\ell$ is a partition of $X \times Y$ into disjoint subsets. (Figure 6)

**Claim 8.** For any $v$ in protocol $P$, $R_v$ is a rectangle

**Proof.** by induction of depth of $v$:

- Base case: $R_{\text{root}} = X \times Y$ is certainly a rectangle
- Inductive step: Suppose true at node $w$ of $P$ so $R_w = U \times V$. Say Bob speaks at $w : f_w(y) \in \{0, 1\}$. Then there are two possible divisions of the $V$ in $R_w$
  
  - $V_0 = \{y \in V : f_w(y) = 0\}$
  - $V_1 = \{y \in V : f_w(y) = 1\}$

  So $R_v$ is either $U \times V_0$ or $U \times V_1$ at child $v$ of $w$ (Figure 5).

**Fact 9.** If $P$ is a protocol for $f : X \times Y \rightarrow Z$, then for any leaf $\ell$ of $P$, rectangle $R_\ell$ is $f$-monochromatic. That is every $(x, y)$ entry in $R_\ell$ is labeled $f(x, y)$.
The leaves of protocol $P$ for $f$ induce a partition of the $X \times Y$ matrix into $f$-monochromatic rectangles. The number of leaves in $P$ is the number of rectangles in the partition. (Figure 6)

This gives us an ‘in’ for lower bounds. If we can show that every partition of $X \times Y$ into $f$-monochromatic rectangles must use $\geq t$ rectangles, then every protocol for $f$ must have $\geq t$ leaves. Hence the cost $\geq \log t$, so $D(f) \geq \log t$.

**Example 10** (EQ rectangle l.b.). $X = Y = \{0, 1\}^n$, $f = \text{EQ}$. Each row has value ‘1’ only on a single column, the one it is equal to. Therefore each rectangle with label ‘1’ is a single cell and there are $2^n$ many of them. We can likewise show that there must be $2^n$ rectangles for the ‘0’ leaves, so the number of rectangles is $2 \cdot 2^n$ and the cost is $\log(2 \cdot 2^n) = n + 1$. (Figure 7)

Therefore $D(\text{EQ}) = n + 1$.
3.2 Communication Complexity of Relations

Definition 11. A relation $R$ is a subset of $X \times Y \times Z$. Equivalently, we can view $R$ as a $f : X \times Y \rightarrow 2^Z$ ($2^Z$ is all subsets of $Z$.)

A communication problem corresponds to relation $R$: $A$ gets $x \in X$ and $B$ gets $y \in Y$. Both must jointly output a single $z$ which must satisfy $(x, y, z) \in R$. Note that it could be that $x, y$ has no $z$ that satisfies $(x, y, z) \in R$, we say that such a $(x, y)$ is an illegal input.

The communication matrix of a relation is a $X \times Y$ matrix, but each $(x, y)$ entry is $\{z \in Z : (x, y, z) \in R\}$. If an $(x, y)$ in the matrix is an illegal input, its entry is $\emptyset$.

The protocol for a relation is the same as before. Each leaf in protocol $P$ is still labeled with a single $z \in Z$. The protocol computes relation $R$ if for every legal $(x, y)$, the protocol reaches a leaf labeled with a $z$ s.t. $(x, y, z) \in R$.

Similarly, the cost of protocol $P$: $\text{cost}(P) = \text{depth of the tree}$ and we use $D(R)$ to denote the deterministic cc of relation $R$. $D(R)$ is the minimum cost of any protocol $P$ for $R$.

Definition 12. Let $U \subseteq X$ and $V \subseteq Y$ and $R$ be a relation. We say that $U \times V$ is an $R$-monochromatic rectangle if $\exists z \in Z$ s.t. $\forall$ legal pairs $(x, y) \in U \times V$, $(x, y, z) \in R$.

This means that all non-empty entries in such a rectangle must share some $z$, but there could also be $\emptyset$ entries. We can use the same lower bound technique that we had with functions with relations.

Remark 1. Let $\#(R)$ refer to the minimum number of leaves in any protocol for a relation $R$.

Note that the number of leaves in the protocol corresponds exactly to the number of leaves in the boolean formula. (to be shown later)
4 Connection Back to Boolean Functions

Fix $f : \{0, 1\}^n \rightarrow \{0, 1\}$. We’ll consider the following relation:

- $X = f^{-1}(1)$
- $Y = f^{-1}(0)$
- $Z = [n]$
- $R_f = \{(x, y, z) : x \in X, y \in Y, x_z \neq y_z\}$

Alice gets an input that $f$ would label 1, Bob gets an input that $f$ would label 0, they have to find an index $z$ where their inputs disagree. Note that there are no illegal inputs. Since $X$ and $Y$ are disjoint, there must be at least one index in which any $x$ and $y$ disagree.

**Example 13** (PAR function relation). Let $f$ be a PAR function, this means that $X = \{x : \text{PAR}(x) = 1\}$ and $Y = \{y : \text{PAR}(y) = 0\}$.

The protocol for $R_{\text{PAR}}$ is binary search (similar to example 2 above)

- $A$ and $B$ maintain interval $[i, k]$ s.t. $\text{PAR}(x_i, \ldots, x_k) \neq \text{PAR}(y_i, \ldots, y_k)$ (this is our invariant)
- Initialize $[i, k] = [1, n]$
- At each stage:
  1. $A$ and $B$ calculates the midpoint $j$ of $[i, k]$
  2. $A$ sends $\text{PAR}(x_i, \ldots, x_j)$
  3. $B$ compares that with $\text{PAR}(y_i, \ldots, y_j)$, if equal sends 0, else 1
  4. $A$ recurses on the half that maintains the invariant

Again we see that there will be $\log n$ many stages, but each stage only requires 2 bits of information. Therefore the total cost of this protocol is $2 \log n$ and this protocol has $n^2$ many leaves. This is actually optimal for PAR.

Note that this example mirrors the boolean formula we saw in the first lecture for PAR. In that formula we built up a tree of recursive calls to the first and second halves of the input string to determine the parity of the strings.
Definition 14. If \(x, y \in \{0,1\}^n\) are s.t. \(x_i \neq y_i\) on exactly one \(i \in [n]\), \(x, y\) are neighbors

Lemma 15 (Khrapchenko’s l.b. for cc of relations over \(\{0,1\}^n\)). [Khr71] Let \(X, Y\) be disjoint subsets of \(\{0,1\}^n\); \(N = \{(x, y) : x \in X, y \in Y, x \text{ and } y \text{ are neighbors}\}; Z = [n];\) and \(R = \{(x, y, z) : x \in X, y \in Y, x_z \neq y_z\}\). Then

\[
\#(R) \geq \frac{|N|^2}{|X| \cdot |Y|}
\]

Proof. Consider an optimal partition into monochromatic rectangles \(R_1, \ldots, R_t\). We’ll show that \(t \geq \frac{|N|^2}{|X| \cdot |Y|}\)

Let \(m_i\) be the number of elements of \(N\) in \(R_i\), so \(m_i = |N \cap R_i|\). Also let \(|R_i| = \text{number of elements in } R_i\). Therefore we have

\[
|N| = \sum_{i=1}^{t} m_i
\]

\[
|X| \cdot |Y| = \sum_{i=1}^{t} |R_i|
\]

Consider \(R_i\), let \(j\) be an element that is in every cell of \(R_i\). Recall that this means that all the \(x\)’s for the rows in \(R_i\) differ from the \(y\)’s for the columns at the \(j\)th bit (and maybe also elsewhere).

Claim 16. Each row (or column) of \(R_i\) contains at most one neighbor (element of \(N\))

For a fixed row \(x\), the only possible neighbor is where \(y\) is exactly \(x\) with the \(j\)th bit flipped. All other neighbors of \(x\) do not have the \(j\)th bit flipped, so can’t appear in \(R_i\). Similarly for a fixed column \(y\). (Figure 8)
This means the (number of rows of \( R_i \)) \( \geq m_i \) and (number of columns of \( R_i \)) \( \geq m_i \); therefore
\[
|R_i| \geq m_i^2
\]

Using the Cauchy-Schwarz inequality we are able to put these together
\[
|N|^2 = \left( \sum_{i=1}^{t} m_i \right)^2 \leq t \cdot \sum_{i=1}^{t} m_i^2
\]
\[
\leq t \cdot \sum_{i=1}^{t} |R_i|
\]
\[
\leq t \cdot |X| \cdot |Y|
\]

Therefore \( \frac{|N|^2}{|X| \cdot |Y|} \leq t \) \( \blacksquare \)

**Example 17** \((R = R_{PAR})\).

\[
X = \text{PAR}^{-1}(1)
\]
\[
Y = \text{PAR}^{-1}(0)
\]
\[
|X| = |Y| = 2^{n-1}
\]
\[
|N| = |X| \cdot n
\]

For \(|N|\), recall that each bit in an \( x \in X \) can be flipped to get a neighbor \( y \in Y \).

We know that
\[
\frac{|N|^2}{|X| \cdot |Y|} = \frac{(|X| \cdot n)^2}{|X| \cdot |X|} = n^2
\]

Using Khrapchenko’s lemma we have: \( \#(\text{PAR}) \geq n^2 \). Earlier in example 13 we constructed a protocol for this relation which had \( n^2 \) many leaves, so we know that is an upper bound. Therefore we have that the \( \#(\text{PAR}) = n^2 \)

**Example 18** \((R = R_{MAJ})\). Note that if we include the entire space of \( X \) and \( Y \) we grow the denominator of our bound, thus weakening it. So for majority we want to
restrict $X$ and $Y$ to the sets that actually have neighbors in the other.

$$X = \text{inputs with exactly } \frac{n+1}{2} \text{ '1's}$$

$$Y = \text{inputs with exactly } \frac{n-1}{2} \text{ '1's}$$

$$|X| = |Y| = \left( \frac{n}{n+1} \right) \approx \frac{2^n}{\sqrt{n}}$$

$$|N| = |X| \cdot \frac{n+1}{2}$$

For $|N|$, each $x \in X$ can be paired to a neighbor $y \in Y$ by flipping one of the ‘1’s in $x$ to a ‘0’ and each $x$ by definition has $\frac{n+1}{2}$ many ‘1’s. Putting these values into our bound

$$\frac{|N|^2}{|X| \cdot |Y|} = \frac{|X|^2 \cdot \left( \frac{n+1}{2} \right)^2}{|X| \cdot |X|} = \left( \frac{n+1}{2} \right)^2$$

Again using Khrapchenko’s lemma we have that $\#(R_{maj}) \geq \Omega(n^2)$

(HW) Show Khrapchenko’s lemma can never give a lower bound $\#(R) = \omega(n^2)$.

References

