1 Introduction

1.1 Last time

- Finished Valiant ’84 [4]
  Valiant proved in 1984 that MAJ has an $O(n^{5.3})$-size monotone formula using (randomized) construction of small-depth monotone formulas for majority.

- Communication Complexity for functions and relations.

  Lemma 1. (Khrapchenko’s l.b. for cc of relations over $\{0,1\}^n$) [3]. Let $X,Y$ be disjoint subsets of $\{0,1\}^n$; $N = \{(x,y) : x \in X, y \in Y, x$ and $y$ are neighbors$\}$; $Z = [n]$, and $R = \{(x,y,z) : x \in X, y \in Y, x \neq y\}$. Then

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

  If we treat $X = f^{-1}(1)$ and $Y = f^{-1}(0)$ for a function $f : \{0,1\}^n \rightarrow \{0,1\}$, it means $\#(R_f) \leq \frac{|N|^2}{|X| \cdot |Y|}$. The function $f$ could be MAJ, PAR, etc.

1.2 Today

- Correspondence between the communication protocol for $R_f$ and formula for $f$.
  (We can get formula l.b. from CC l.b.s!)

- Application: $\text{msize}(CON) = n^{\Omega(\log n)}$ (actually $\text{msize}(CON) = n^{\Theta(\log n)}$)
2 Correspondence

Key insight for correspondence Protocol for $R_f$ can be seen as a machine for separating $X = f^{-1}(1)$ from $Y = f^{-1}(0)$. Actually, the formulas do the same thing! (works by Karchmer/Wigderson [2], Yannakakis)

2.1 EXACT CORRESPONDENCE

<table>
<thead>
<tr>
<th>communication protocol $P$ for $R_f$</th>
<th>formula $F$ for $f$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\circ$ Internal node of $P$</td>
<td>$\leftrightarrow$ $\circ$ gate of $F$</td>
</tr>
<tr>
<td>• Alice node</td>
<td>$\leftrightarrow$ $\bullet$ $\lor$-gate</td>
</tr>
<tr>
<td>• Bob node</td>
<td>$\leftrightarrow$ $\bullet$ $\land$-gate</td>
</tr>
<tr>
<td>$\circ$ #leaves in $P$</td>
<td>$\leftrightarrow$ $\circ$ size($F$)</td>
</tr>
<tr>
<td>$\circ$ depth of $P$</td>
<td>$\leftrightarrow$ $\circ$ depth($F$)</td>
</tr>
<tr>
<td>$\circ$ leaf of $P$,</td>
<td>$\leftrightarrow$ $\circ$ leaf of $F$ labeled with $x_i$</td>
</tr>
<tr>
<td>labeled $i \in [n]$, s.t.</td>
<td></td>
</tr>
<tr>
<td>$\forall (x, y)$ correspond to this leaf, $x_i = 1, y_i = 0$</td>
<td>$\forall (x, y)$ correspond to this leaf, $x_i = 0, y_i = 1$</td>
</tr>
</tbody>
</table>

2.2 Correctness

Lemma 2. ($F \rightarrow P$) Given formula $F$ for Boolean function $f$, viewing $F$ as protocol $P$ via EC(Exact Correspondence), have that $P$ computes $R_f$.

Proof. Recall Alice has $x \in X = f^{-1}(1)$, Bob has $y \in Y = f^{-1}(0)$.

A, B make their way down from the root of $F$ to a leaf, and maintaining the following invariant:

The function $g$ computed at current gate is s.t. $g(x) = 1, g(y) = 0$.

This way, when reach a leaf, its variable $x_i$ is s.t. $x_i \neq y_i$.

(Using induction top-down)

• Invariant true at root, since root computes $f$.

• Suppose current node is $\lor$, and the current function at the node is $g$. Denote the child node with 0-move as $g_0$, and the child node with 1-move as $g_1$, i.e., $g = g_0 \lor g_1$. 
Since \( g(y) = g_0(y) \lor g_1(y) = 0 \), there is \( g_0(y) = 0 \) and \( g_1(y) = 0 \). Then, no matter which bit \( b \) Alice sends to Bob, it will be true that \( g_b(y) = 0 \) which preserves this part of the invariant.

Since \( g(x) = g_0(x) \lor g_1(x) = 1 \), there is \( g_0(x) = 1 \) for some \( b \in \{0, 1\} \). Alice sends \( b \) to Bob and they proceed to \( g_b \), preserving the other part of the invariant, that \( g_b(x) = 1 \).

- Suppose current node is \( \land \), and the current function at the node is \( g \). Denote the child node with 0-move as \( g_0 \), and the child node with 1-move as \( g_1 \), i.e.,
\[
g = g_0 \land g_1.
\]

Since \( g(x) = g_0(x) \land g_1(x) = 1 \), there is \( g_0(x) = 1 \) and \( g_1(x) = 1 \). Then, no matter which bit \( b \) Bob sends to Alice, it will be true that \( g_b(x) = 1 \) which preserves this part of the invariant.

Since \( g(y) = g_0(y) \land g_1(y) = 0 \), there is \( g_b(y) = 0 \) for some \( b \in \{0, 1\} \). Bob sends \( b \) to Alice and they proceed to \( g_b \), preserving the other part of the invariant, that \( g_b(y) = 0 \).

At the leaf node \( g \) outputs some \( x_i \) or its negation. By the invariant \( g(x) \neq g(y) \Rightarrow x_i \neq y_i \) and our protocol returns \( i \). Therefore, \( P \) computes \( R_f \).

\[ \square \]

**Corollary 3.** From Lemma 2, we have, \( \forall \) boolean function \( f : \)
\[
\#(R_f) \leq \text{size}(f)
\]
\[
D(R_f) \leq \text{depth}(f)
\]

So, \( \text{size}(\text{PAR}/\text{MAJ}) = \Omega(n^2) \) (by Khrapchenko [3])

**Lemma 4.** \((P \rightarrow F)\) Given protocol \( P \) for \( R_f \), viewing it as formula \( F \) via EC, have that \( F \) computes \( f \).

**Proof.** Recall the \( P \rightarrow F \) conversion:

- \( A \text{ - node} \rightarrow \lor \)
- \( B \text{ - node} \rightarrow \land \)
- For leaf node \( l \) with \( i, \text{s.t.} \) the corresponding rectangle \( U \times V \) has \( x_i = 1, y_i = 0, \forall(x,y) \in U \times V \), label the literal \( x_i \)
• For leaf node \( l \) with \( i, \) s.t. the corresponding rectangle \( U \times V \) has \( x_i = 0, y_i = 1, \forall (x, y) \in U \times V, \) label the literal \( x_i \).

Then, we show the invariant that for each gate, function \( g \) computed at that gate has, \( \forall x \in U, \forall y \in V : g(x) = 1, g(y) = 0. \) Here \( U \times V \) is the set of inputs reaching that gate.

Notice that, at the root node, \( U \times V \) is \( X \times Y = f^{-1}(1) \times f^{-1}(0). \) So, if the invariant holds at the root, the function \( g \) computed there is 1 on every \( x \in f^{-1}(1) \) and 0 on every \( x \in f^{-1}(0) \) and the protocol computes \( f. \)

(Using induction bottom-up)

• At the leaf node, It is true directly by EC.

• Consider internal node of the protocol tree. Let \( g \) be the function computed by that node, and let \( U \times V \) be set of inputs reaching \( g. \) We prove the induction step by supposing the current node is A-node, while the proof of B-node is symmetric.

A’s function at the node will split \( U \) to two disjoint part, denoted as \( U_0, U_1. \) The \( U_0 \times V \) is the set of inputs reaching the child node with 0-move, and the \( U_1 \times V \) is the set of inputs reaching the child node with 1-move. Also, let \( g_0 \) denotes the function at the child node with 0-move, and \( g_1 \) denotes the function at the child node with 1-move.

Since \( g = g_0 \lor g_1, \) with the inductive hypothesis, there is,
\[
\forall x \in U, g(x) = g_0(x) \lor g_1(x) = 1 \\
\text{(Because, there must exits } b \in \{0, 1\}, x \in U_b, \text{ so } g_b(x) = 1) \\
\forall y \in V, g(y) = g_0(y) \lor g_1(y) = 0
\]

So, invariant holds at \( g. \)

Therefore, \( F \) computes \( f. \) 

Corollary 5.

\[
\text{size}(f) = \#(R_f) \\
\text{depth}(f) = D(R_f)
\]

So, to prove l.b.s for boolean formula of \( f, \) “only” have to prove l.b. of CC of \( R_f. \) This point of view has led to powerful l.b.s!

E.g. \( \text{msize}(\text{CON}) = n^\Omega(\log n). \) We will prove it in next section.
3 Depth and size of monotone function

What about the size and depth of monotone function formula?

**Definition 6.** Let function $f : \{0, 1\}^n \to \{0, 1\}$ be monotone, Let $X = f^{-1}(1), Y = f^{-1}(0)$, define

$$M_f = \{(x, y, i) : x \in X, y \in Y, s.t. x_i = 1, y_i = 0\}$$

Notice: EC holds as before (just erase the the line for $x_i = 0, y_i = 1$)

So, we can get l.b. on msize($f$), mdepth($f$) by lower bounding #($M_f$), $D(M_f)$.

We’ll do this for $f = CON$. (Connectivity function)

**Definition 7.** The connectivity function is defined as bellow:

$$CON : \{0, 1\}^{n^2} \to \{0, 1\}$$

Input is a 0/1 matrix, which is the adjacent matrix of a directed graph $G$.

Output is an indicator of whether there is a directed path in $G$ from $s = 1$ to $t = n$. If there is, the output is 1, otherwise is 0.

$CON$ is a monotone function, since adding edges can only change the output from 0 to 1.

Q: What’s mdepth($CON$)?

A: $\Omega(\log^2 n)$ (prove by depth reduction)

– for monotone formulas, gives #($M_{CON}$) = $n^{\Theta(\log n)}$

Next, we will prove it in two parts, the easy part is proving the upper bound, and the hard part is proving the lower bound.

3.1 Upper bound (easy part)

**Claim 8.** $mdepth(CON) = O(\log^2 n)$

*Proof.* Main technique: “repeated squaring”.

The input to $CON$ for graph $G$ is the adjacent matrix for $G$, s.t.

$$G_{i,j} = \begin{cases} 
1 & \text{if } \exists i \to j \text{ edge} \\
0 & \text{o.w.}
\end{cases}$$
Then we construct $H$ be the graph $g$ with self-loops, i.e. $\forall i \in [n], G_{i,i} = 1$.

$$H_{i,j} = \begin{cases} 1 & \text{if } \exists i \sim j \text{ path of length } \leq 1 \\ 0 & \text{o.w.} \end{cases}$$

Then, we do “Boolean squaring” to the matrix $H$:

$$H^2_{i,j} = \bigvee_{k=1}^{n} (H_{i,k} \land H_{k,j})$$

Then we get $H^2$ as bellow:

$$H^2_{i,j} = \begin{cases} 1 & \text{if } \exists i \sim j \text{ path of length } \leq 2 \\ 0 & \text{o.w.} \end{cases}$$

Notice that $H^2_{i,j}$ is $O(\log n)$-depth and $\text{poly}(n)$-size formula in $\text{H}_{i,j}$ variables.

Squaring $H^2$ get $H^4$ in the same way, we can get an indicator matrix for paths with length $\leq 4$.

After $\log n$ times squaring, we get $H^n$ which indicate for paths with length $\leq n$. Since if there is a path from $s = 1$ to $t = n$, there must exists such a path with length $\leq n$, the function $H^n_{i,n}$ is actually a function for connectivity. Formally, $H^n_{i,n} = 1$ iff $\exists$ path from $s = 1$ to $t = n$.

Because there are $\log n$ times of squaring, the depth of $H^n_{i,n}$ is upper bounded by $O(\log n) \times \log n$, which means $\text{mdepth}(\text{CON}) = O(\log^2 n)$

\[\blacksquare\]

### 3.2 Lower bound(hard part)

(Karchmer/Wigderson [2], Grigni/Sipser [1])

**Theorem 9.** $\text{mdepth}(\text{CON}) = \Omega(\log^2 n)$

We prove it by showing $D(M_{\text{CON}}) = \Omega(\log^2 n)$

Consider the protocol of $\text{CON}$:

A gets element $x \in X = \{\text{all graphs has } 1 \sim n \text{ path}\}$

B gets element $y \in Y = \{\text{all graphs has no } 1 \sim n \text{ path}\}$

Must output an edge that in $x$, not in $y$. 
1-st Reduction  restricted domain(only consider “special” graphs):

Consider only layered graphs, which has $n$ nodes and $l + 2$ layers indexed as $0, 1, ..., l + 1$. Each layer has width $w$, that there is $(l + 2)w = n$. (For example, see figure 1)

Every edge goes from some layer $i$ to next layer $i + 1$.

Define

$$X' = \{\text{all layered graphs has a path from 1 to } n\}$$

$$Y' = \{\text{all layered graphs has no path from 1 to } n\}$$

Then, the new relation $M$ is same with $M_{CON}$ but restricted to $X', Y'$.

Since $M \subseteq M_{CON}$, there is $D(M_{CON}) \geq D(M)$. So it is sufficient to show $D(M) = \Omega(\log^2 n)$

2-nd Reduction  (Reduce to “FORK” relation)

**Definition 10.** $FORK$ is a relation over $X \times Y \times Z$, where $X = Y = [w]$, $Z = \{0, 1, ..., l\}$.

$$FORK = \{(x, y, i) \in X \times Y \times Z : x = x_1, ..., x_l, y = y_1, ..., y_l, x_i = y_i \land x_{i+1} \neq y_{i+1}\}$$

Think of $x, y$ as having: $x_0 = y_0 = 1, x_{i+1} = w, y_{i+1} = w - 1$

Then, $FORK$ has no illegal inputs.

**Claim 11.** $D(M) \geq D(FORK)$

- Then it is sufficient to show $D(FORK) = \Omega(\log^2 n)$

**Proof.** Given protocol for $M$, can solve FORK as follows:(See figure 1.)

A: Hold the FORK input $x = x_1, ..., x_l$.

Construct graph $G_x$: As before, it has $n$ nodes, $l + 2$ layers and each layer with width $w$. Then view $x$ as a path from node 1 to $n$ in the layered graph. In layer $i$, path goes through $x_i$. Only add this path to graph $G_x$.

Notice: the path is start at node 1 in layer 0, and ends at node $n$ in layer $l + 1$.)

B: Hold the FORK input $y = y_1, ..., y_l$.

Construct graph $G_y$: As before, it has $n$ nodes, $l + 2$ layers and each layer with width $w$. Then view $y$ as a path from node 1 to $n - 1$ in the layered graph (Suppose node $n - 1$ is the $(w - 1)$-th node at the last layer). Add this path to graph $G_y$. 
Additionally, for every node \( u \) that is not on the path \( y \), add edges from \( u \) to all nodes in the next layer.

\( (G_y \) missing exactly those edges that leave his \( y \)-path).

Then, \( G_x \) is a graph has a 1 \( \sim \) \( n \) path, while \( G_y \) has no 1 \( \sim \) \( n \) path. So, \( G_x \in X' \), \( G_y \in Y' \). The protocol for \( M \) will output an edge that in \( G_x \) but not in \( G_y \). Since every edge from a node that not in path \( y \) are in \( G_y \), the output must be an edge in path \( x \) and from a node in path \( y \) but to a different node in \( y \), which exactly the answer for FORK.

Then, our goal is to prove \( D(\text{FORK}) = \Omega(\log^2 n) \).

\( \square \)

we’ll show it’s \( \Omega(\log l \log w) \), which imply \( \Omega(\log^2 n) \).

Note: There is a \( O(\log l \log w) \) protocol for \( \text{FORK} \).(By binary search, using \( \log l \) stages for layer searching, and \( \log w \) time per stage)

The idea of the l.b. proof is: “amplifying” accuracy of somewhat good protocols.

**Definition 12.** (Key definition)
For $0 \leq \alpha \leq 1$ and $l \geq 1$, a protocol $P$ is an $(\alpha, l)$-protocol for FORK, if:

$$\exists S \subseteq [w]^l, |S| \geq \alpha w^l, \text{s.t.}\forall x, y \in S, \text{ } P \text{ succeeds on } (x, y)$$

Specially, $(1, l)$-protocol works correctly on all inputs.

In order to prove our goal, we first introduce 3 lemmas. Two of them are easy, and the hard one will be proved in next lecture.

**Lemma 13. (Easy)** For $c \geq 1$, if there is a $c$-bit $(\alpha, l)$-protocol, then there’s a $(c-1)$-bit $(\frac{\alpha}{2}, l)$-protocol.

- Can save 1 bit at cost of halving set it works for.

**Proof.** Without loss of generality, suppose $A$ speaks first. And $S$ is the set satisfies the requirement, where $|S| \geq \alpha w^l$.

Let $S_0$ be the set of inputs where $A$ says 0. And $S_1$ be the set of inputs where $B$ says 1.

There is a $b \in \{0, 1\}$, such that $|S_b| \geq \frac{|S|}{2}$.

Construct the protocol from the $c$-bit protocol for $S_b$ as disregard the first speak and both carry on as if $A$ said $b$. Then the protocol is a $(c-1)$-bit $(\frac{\alpha}{2}, l)$-protocol. ■

**Lemma 14. (Easy)** If $\alpha > \frac{1}{w}$, then $c(\alpha, l) \geq 1$. The $c(\alpha, l)$ means the minimum number of bits used by any $(\alpha, l)$-protocol.

- Any protocol that works for $> \frac{1}{w}$ fraction of all inputs must use $> 0$ bits.

**Proof.** (by contradiction)

Suppose $P$ is the $(\alpha, l)$-protocol using zero bits.

Since there is no communication, they both output $z = l$. If not, the protocol will wrong on any $x \in S, x = y$.

Then, whatever $A$’s string is in coordinate $z = l$, only $\frac{1}{w}$ fraction of $[w]^l$ agrees there.(Because they must agree on the $l$-th bit).

So, there is $|S| \leq \frac{1}{w}$, which makes a contradiction. ■

**Lemma 15. (key lemma)** let $\alpha \geq \frac{12}{w}$. If there’s a $c$-bit $(\alpha, l)$-protocol for FORK, then there’s a $c$-bit $(\frac{\sqrt{\alpha}}{2}, \frac{l}{2})$-protocol for FORK.

- a “good” protocol for length $l$ strings gives an “even better” algorithm for shorter (length $l/2$) strings.
**Proof using 3 lemmas** We’ll prove the key lemma in next lecture. Next, we prove that $D(FORK) = \Omega(\log w \log l)$.

To show $c(1,l) \geq \Omega(\log l \log w)$, we show $c\left(\frac{1/2}{\sqrt{w}}, l\right) \geq \Omega(\log l \log w)$.

Given $\left(\frac{1/2}{w^{1/3}}, l\right)$-protocol: By lemma 13, used $(\frac{1}{3} \log(w) - 1)$ many times, we can save $\frac{1}{3} \log w - 1$ bits, then get a $\left(\frac{1}{w^{2/3}}, l\right)$-protocol.

So, combine with lemma 14, there is

$$c\left(\frac{1/2}{w^{1/3}}, l\right) \geq \Omega(\log w) + c\left(\frac{1}{w^{2/3}}, l\right)$$

Given $(\frac{1}{w^{2/3}}, l)$-protocol, by lemma 15, there is a $\left(\frac{1/2}{w^{1/3}}, \frac{l}{2}\right)$-protocol.

Now, there is:

$$c\left(\frac{1/2}{w^{1/3}}, l\right) \geq \Omega(\log w) + c\left(\frac{1/2}{w^{1/3}}, \frac{l}{2}\right)$$

Repeat the last step $\frac{1}{2} \log l = \log \sqrt{l}$ times, there is:

$$c\left(\frac{1/2}{w^{1/3}}, l\right) \geq \Omega(\log w \log l) + c\left(\frac{1/2}{w^{2/3}}, \sqrt{l}\right)$$

Conclude that $D(FORK) = \Omega(\log w \log l)$

**Preliminaries for proving key lemma** The inputs is in $[w]^l$. Consider $[w]^l$ as a matrix, denoted as $M = [w]^{l/2} \times [w]^{l/2}$. The rows is corresponding to the prefix of inputs, and the columns is corresponding to the suffix of inputs.

For $S$ mentioned in definition 12, which satisfies $|S| \geq \alpha \cdot w^l$. We define $M$ as an 0/1-matrix, where $M_{u,v} = 1$ iff $uv \in S$. See figure 2.

Notice that $M$ is $\alpha$-dense with 1’s.

**Fact 16.** Let $M$ be $r \times r$ 0/1-matrix with $\geq \alpha$ fraction of 1-entries. Then either

1. some row has $\geq \sqrt{\frac{\alpha}{2}}$ fraction of 1’s, or

2. at least $\sqrt{\frac{\alpha}{2}}$ fraction of rows that each have $\geq \frac{\alpha}{2}$ fraction of 1’s.
Figure 2: \([w]^{l/2} \times [w]^{l/2}\) 0/1 matrix \(M\). \(S\) is the subset of \(M\) which is comprised of all 1-entries. There is \(|S| \geq \alpha |M|\).

**Proof.** (by contradiction)  
If 1, 2 both false, then  

\[
\text{tot fraction of 1's} = \text{contribution from rows with } < \frac{\alpha}{2} \text{ fraction of 1's} \\
+ \text{contribution from rows with } \geq \frac{\alpha}{2} \text{ fraction of 1's} \\
< \frac{\alpha}{2} + \sqrt{\frac{\alpha}{2}} \sqrt{\frac{\alpha}{2}} = \alpha
\]

Which makes a contradiction. ■

**References**

