Communication Complexity 29 Aug, 2011 (@ IMSc)

2. Depth lower bound for matching

Lecturer: Prahladh Harsha *Scribe: Nitin Saurabh*

In today's lecture, we will first recall the fooling set argument we discussed in the previous lecture and then explore a connection between communication complexity and formula depth. The references for today's lecture include Sections 1.3 and 10.1–10.3 of Kushilevitz and Nisan's book on Communication Complexity [KN97].

2.1 Rectangles

Definition 2.1. *Let $f : X \times Y \to V$. A subset R of $X \times Y$ is a* rectangle[1] *if it is of the form $A \times B$ for some $A \subseteq X$ and $B \subseteq Y$. The rectangle R is said to be* monochromatic *(wrt. f) if f is constant on R. A monochromatic rectangle R is a 0-rectangle if $f(R) = \{0\}$; it is a 1-rectangle if $f(R) = \{1\}$.*

Observation 2.2. *A subset S of $X \times Y$ is a rectangle iff for all $x, x' \in X$ and $y, y' \in Y$ $(x, x') \in S$ and $(y, y') \in S$ implies $(x, y') \in S$ and $(x', y) \in S$.*

Observation 2.3. *Any deterministic protocol $\mathcal{P}$ on $X \times Y$ induces a partition of $X \times Y$. If $\mathcal{P}$ computes a function $f : X \times Y \to \{0, 1\}$, then the rectangles of this partition are 0- and 1-rectangles. There is one such rectangle for each leaf of P. Similarly, a non-deterministic protocol for f induces a set of 1-rectangles of f whose union (and not necessarily a partition) is $f^{-1}(1)$.*

2.1.1 The fooling set argument

Definition 2.4. *Let $f : X \times Y \to \{0, 1\}$. A set $S \subset X \times Y$ is called a* fooling set *for f if no monochromatic rectangle wrt. f contains more than one element of S.*

Lemma 2.5. *Let S be a fooling set for f. Then, $\mathrm{D}(f) \geq \log_2 |S|$. If $S \subseteq f^{-1}(1)$, then $\mathrm{N}(f) \geq \log_2 |S|$.*

Proof. Since no two elements of S can be in the same rectangle, Observation 2.3 implies that any deterministic protocol tree for f must have at least $|S|$ leaves and, therefore, depth at least $\log_2 |S|$. The lower bound on $\mathrm{N}(f)$ is similar. □

Exercise: Exhibit fooling sets of appropriate sizes for EQ_n and DISJ_n, and conclude tight lower bounds for deterministic and non-deterministic complexities of EQ_n and DISJ_n.

Remark 2.6. *The fooling set argument works by showing that many rectangles are necessary because any one rectangle can cover only a small part of the fooling set. In general, if there is a probability distribution $\mu : X \times Y \to [0, 1]$ such that $\sum \mu(x, y) = 1$, such that for ever*

[1] also called combinatorial rectangle

rectangle R, $\mu(R) \le \delta$, then $\mathrm{D}(f) \ge \log_2(1/\delta)$. The fooling set argument is just a special case of this where μ is the uniform distribution supported on the fooling set. A similar conclusion holds for non-deterministic complexity when the support of μ is contained in $f^{-1}(1)$.

2.2 Communication complexity and formula depth

Definition 2.7. *A relation is subset F of $X \times Y \times I$ (for some sets X, Y and I), where for each $(x, y) \in X \times Y$, there is an $i \in I$ such that $(x, y, i) \in F$. The communication complexity of such a relation is defined in the following natural way: Alice is given an $x \in X$, and Bob a $y \in Y$; they must determine an $i \in I$ such that $(x, y, i) \in F$. Corresponding to the Boolean function $f : \{0,1\}^n \times \{0,1\}$, we have its Karchmer-Wigderson relation $\mathsf{KW}_f \subseteq f^{-1}(1) \times f^{-1}(0) \times [n]$ defined by $\mathsf{KW}_f = \{(x, y, i) : x_i \neq y_i\}$.*

Definition 2.8. *A formula on input variables $z_1, z_2, \ldots, z_n$ is circuit of the following form. The underlying graph of a formula is a tree, where internal nodes have labelled $\vee$ or $\wedge$, and each leaf is labelled by a literal of the form x_i or $\neg x_i$. The depth of a formula is the length of the longest path from a root to an input. Let $\mathrm{depth}(f)$ be the minimum depth of a formula computing f.*

Theorem 2.9 ([KPPY84, KW90])**.** *For a Boolean function $f : \{0,1\}^n \to \{0,1\}$, $\mathrm{D}(\mathsf{KW}_f) = \mathrm{depth}(f)$.*

Claim 2.10. $\mathrm{D}(\mathsf{KW}_f) \le \mathrm{depth}(f)$

Proof. Let F be a formula of depth $\mathrm{depth}(f)$ on variables $z_1, z_2, \ldots, z_n$ computing f. In the protocol, Alice and Bob both travel down F, starting at the root, in order to identify a coordinate where their inputs differ. At all times, they maintain the invariant that the function computed at the current node evaluates to 1 on x (Alice's input) and evaluates to 0 on y (Bob's input). Note that this is true at the beginning as the first node is the root of F.

If the current node is an OR gate then it is Alice's turn to speak. Let the function computed at the current node be $f_0 \vee f_1$. Hence, $f_0(x) \vee f_1(x) = 1$ and $f_0(y) \vee f_1(y) = 0$. Therefore, either $f_0(x) = 1$ or $f_1(x) = 1$. Alice sends a single bit indicating which child evaluates to 1, and they both move to the corresponding node. Note that this node satisfies the invariant. If the current node is labelled $\wedge$, Bob can send a bit indicating which child they must move to next. If the current node is a leaf and its z_i or $\neg z_i$, the protocol returns the value i. The number of bits exchanged in the protocol is the depth of the leaf reached, justifying Claim 2.10. □

Claim 2.11. $\mathrm{D}(\mathsf{KW}_f) \ge \mathrm{depth}(f)$.

Proof. Let $\mathcal{P}$ be the optimal protocol for $\mathrm{D}(\mathsf{KW}_f)$. We will convert its protocol tree into a formula as follows: each internal node in which Alice sends a bit is labelled by $\vee$ and each internal node in which Bob sends a bit is labelled by $\wedge$. Each leaf of the protocol tree is labelled by an index in [n]. Let $S \times T \subseteq f^{-1}(1) \times f^{-1}(0)$ be a set of inputs that lead to a leaf labelled $i \in [n]$. Then either (1) $\forall x \in S$, $x_i = 1$ and $\forall y \in T$, $y_i = 0$; or (2) $\forall x \in S$,

$x_i = 0$ and $\forall y \in T$, $y_i = 1$. To see this, suppose not, then there exist $(x, y) \in S \times T$ such that $x_i = 1$ and $y_i = 0$, and $(x', y') \in S \times T$ such that $x'_i = 0$ and $y'_i = 1$. This implies, $(x, y') \in S \times T$ and $(x', y) \in S \times T$. But that implies that the protocol tree outputs i on these inputs, which is a contradiction. So, in the first case we label the leaf node by z_i, and in the second case we label the leaf node by $\neg z_i$.

Clearly, the depth of the formula is $\mathrm{D}(\mathsf{KW}_f)$. Therefore, it remains to prove that the constructed formula computes f. It is sufficient to prove that for every node of the formula, the function f' corresponding to that node satisfies $f'(x) = 1$ for all $x \in A$ and $f'(z) = 0$ for all $z \in B$, where $A \times B$ are the inputs that reach that node of the protocol tree. This immediately implies that the function computed by the output node is f, because it is 1 for all inputs in $f^{-1}(1)$ and 0 for all inputs in $f^{-1}(0)$.

To justify the claim we use induction on the depth of the formula. Base case: the claim holds for the input nodes by construction. Assume the claim holds for the children v_0 and v_1 of a certain node v. We will now show that it must then hold for the node v itself. Let f_0 and f_1 be the corresponding formulas rooted at nodes v_0 and v_1. Let f' be a function computed at a node v. Let $A \times B$ be the inputs reaching v in the protocol tree. Let us assume that Alice sends Bob a bit in this node. Alice's bit partitions A into A_0 and A_1 such that $A_0 \times B$ is the set of input that reaches v_0 and $A_1 \times B$ reaches v_1. By the induction hypothesis, $\forall y \in B,\ f_0(y) = f_1(y) = 0$, and $\forall x \in A_0,\ f_0(x) = 0$ and $\forall x \in A_1,\ f_1(x) = 1$. Since $f = f_0 \vee f_1$, $\forall y \in B,\ f(y) = 0$ and $\forall x \in A,\ f(x) = 1$. A similar argument applies when it is Bob's turn to communicate when the protocol reaches node v. □

One can also restrict attention to monotone functions.

Definition 2.12. *For $x, y \in \{0,1\}^n$ we say that $x \le y$ if for all i, $x_i \le y_i$. A boolean function $f : \{0,1\}^n \to \{0,1\}$ is called* monotone *if $x \le y$ implies $f(x) \le f(y)$. Let* $\mathrm{depth}^+(f)$ *be the minimum depth of a monotone formula computing f.*

Definition 2.13. *For a monotone Boolean function f on n inputs, let $\mathsf{KW}^+_f = \{(x, y, i) : x \in f^{-1}(1), y \in f^{-1}(0), 1 = x_i \ne y_i = 0\}$.*

Theorem 2.14. *For every monotone boolean function f, $\mathrm{D}(\mathsf{KW}^+_f) = \mathrm{depth}^+(f)$.*

Proof. Same as above. □

2.2.1 Lower bound for matching

Definition 2.15. Match_n *is a function on $\binom{n}{2}$ variables, where the input (in $\{0,1\}^{\binom{n}{2}}$) is interpreted as the characteristic vector of the edge set of a graph G. The value of the function is 1 iff G has a perfect matching. Note that* Match_n *is a monotone function.*

Exercise: Show that Match_n can be computed by an OR-AND-OR monotone formula (unbounded fanin) formula of size $2^{O(n)}$.

Theorem 2.16 ([RW92])**.** $\mathrm{depth}^+(\mathsf{Match}_n) = \Omega(n)$, *where n is the number of vertices.*

Proof. We will show a randomized reduction from DISJ_n to $\mathsf{Match}_{n'}$ (for some $n' = O(n)$) and conclude from Theorem 2.14 that $\mathrm{depth}^+(\mathsf{Match}_{n'}) = \mathrm{D}(\mathsf{KW}^+_f) = \mathrm{R}(\mathsf{DISJ}_n) = \Omega(n) = \Omega(n')$. In the above sequence, we use the result (to be shown later in the course) that $\mathrm{R}(\mathsf{DISJ}_n) = \Omega(n)$.

Disjointness reduces to a covering problem: Consider an instance of DISJ_n. Alice gets $X \subseteq [n]$ and Bob gets $Y \subseteq [n]$. Alice makes the following graph on $3n$ vertices: $V = \{a_i, b_i, c_i : i \in [n]\}$ and edges are added as follows, if $i \in X$, then (a_i, b_i) is an edge else (b_i, c_i) is an edge. So, Alice's graph is a matching of size n on $3n$ vertices. Bob gets a subset T of vertices, where if $i \notin Y$ then $b_i \in T$ else $c_i \in T$ (see Figure 1). Observe that X and Y are disjoint if and only if T covers the edges of Alice's graph.

Figure 1: Alice's graph and Bob's set

Disjointness reduces to $\mathsf{KW}_{\mathsf{Match}(n,3n)}$: The function $\mathsf{Match}(n, 3n)$ (on an input of size $\binom{3n}{2}$) is similar to Match_n, and is 1 iff the graph has a matching of size n. We will show a randomized reduction from the covering problem above to $\mathsf{KW}_{\mathsf{Match}(n,3n)}$: that is Alice will be given a graph with a matching of size $3n$ and Bob a graph with no such matching. Their goal is to determine an edge in Alice's graph that is missing from Bob's.

Alice's input is the same graph as the one in the reduction above. We need to turn Bob's set T into a graph that has no matching of size n. We do this as follows. Bob randomly picks a vertex $v \in T$ and considers $T' = T \setminus \{v\}$. Bob's input for $\mathsf{KW}_{\mathsf{Match}(n,3n)}$ is the complete bipartite graph between T' and $\{a_i, b_i, c_i : i \in [n]\} \setminus T'$. (See Figure 2.)

Alice and Bob use shared randomness to apply a common random permutation σ to the vertices. Let the resulting graphs be G_A and G_B. Suppose Alice and Bob run the protocol for $\mathsf{KW}_{\mathsf{Match}(n,3n)}$ on the input (G_A, G_B) and the protocol returns $e = \{x, y\} \in E(G_A) \setminus E(G_B)$. If e is incident on $\sigma(v)$ (where v is the vertex that was removed from T to produce T'), then Bob declares that T is a cover, otherwise he says that T is not a cover.

Error analysis: Notice that whenever T covers the edges of G_A, Bob does declare it to be so. If T does not cover the edges of G_A, there are at least two edges in Alice's matching that are not in Bob's graph; since the vertices are being permuted randomly each of these edges is equally likely to be picked by the protocol for $\mathsf{KW}_{\mathsf{Match}(n,3n)}$. Thus, with probability at least $\frac{1}{2}$, Bob declares that T does not cover the edges of G_A. We can repeat the protocol to reduce error.

$\mathsf{KW}_{\mathsf{Match}(n,3n)}$ reduces to $\mathsf{KW}_{\mathsf{Match}_{4n}}$: Alice and Bob add n vertices (with identical names) to their graphs and connect them to all the other vertices.

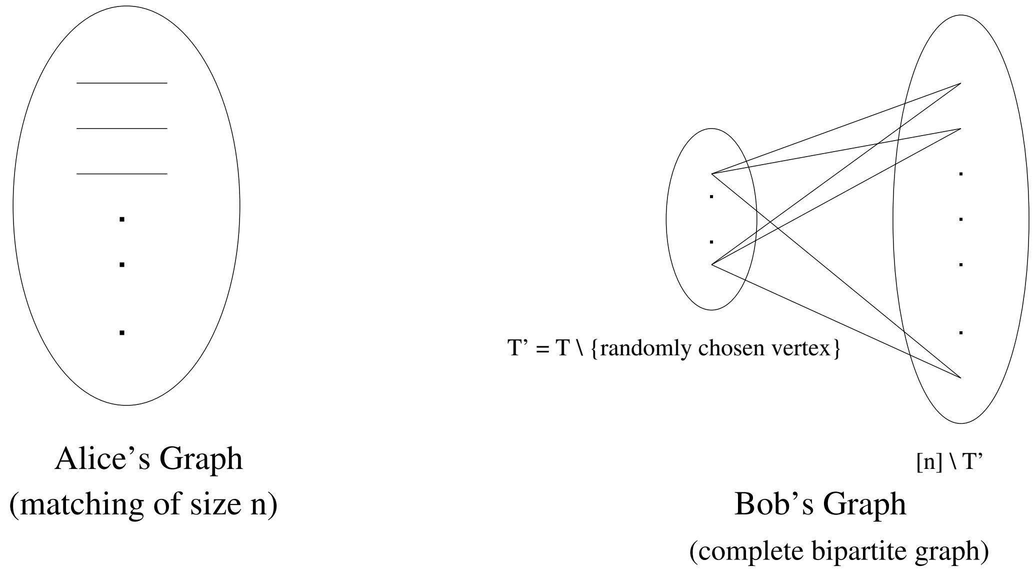

Figure 2: Disjointness to graphs

Clearly, Alice's graph has a perfect matching and Bob's hasn't. A protocol for $\mathsf{KW}_{\mathsf{Match}_{4n}}$ must discover an edge $E(G_A) \setminus E(G_B)$, for all other (newly added) edges are the same in the two graphs. $\square$

References

[KN97] Eyal Kushilevitz and Noam Nisan. *Communication Complexity.* Cambridge University Press, 1997. doi:10.2277/052102983X.

[KPPY84] Maria M. Klawe, Wolfgang J. Paul, Nicholas Pippenger, and Mihalis Yannakakis. *On monotone formulae with restricted depth.* In *Proc. 16th ACM Symp. on Theory of Computing (STOC)*, pages 480–487. 1984. doi:10.1145/800057.808717.

[KW90] Mauricio Karchmer and Avi Wigderson. *Monotone circuits for connectivity require super-logarithmic depth.* SIAM J. Discrete Math., 3(2):255–265, 1990. (Preliminary version in *20th STOC*, 1988). doi:10.1137/0403021.

[RW92] Ran Raz and Avi Wigderson. *Monotone circuits for matching require linear depth.* J. ACM, 39(3):736–744, 1992. (Preliminary version in *22nd STOC*, 1990). doi:10.1145/146637.146684.

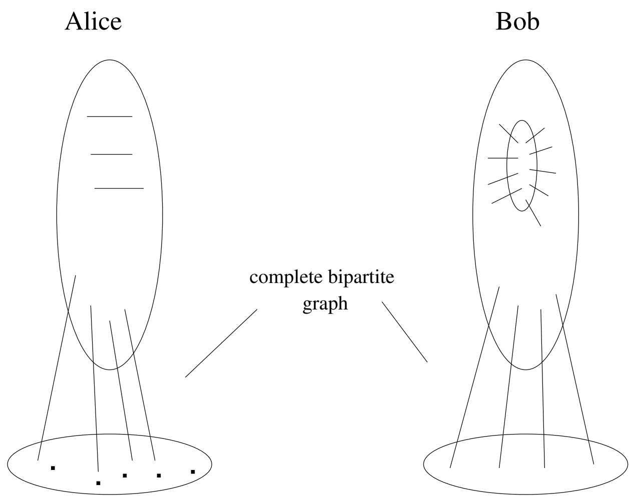

Figure 3: $\mathsf{Match}(n, 3n)$ to Match_{4n}