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. 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 $P$ on $X \times Y$ induces a partition of $X \times Y$. If $P$ 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 $S \subseteq X \times Y$ 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, $D(f) \geq \log_2 |S|$. If $S \subseteq f^{-1}(1)$, then $N(f) \geq \log_2 |S|$.

**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 $N(f)$ is similar.

**Exercise:** Exhibit fooling sets of appropriate sizes for $\text{EQ}_n$ and $\text{DISJ}_n$, and conclude tight lower bounds for deterministic and non-deterministic complexities of $\text{EQ}_n$ and $\text{DISJ}_n$.

**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 $R$, $\mu(R) \leq \delta$, then $D(f) \geq \log_2(1/\delta)$. The fooling set argument is just a special case of this where $\mu$ is the uniform distribution supported on the fooling set. A similar conclusion holds for non-deterministic complexity when the support of $\mu$ 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 $KW_f \subseteq f^{-1}(1) \times f^{-1}(0) \times [n]$ defined by $KW_f = \{(x, y, i) : x_i \neq y_i\}$.

**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 $\lor$ or $\land$, 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 $depth(f)$ be the minimum depth of a formula computing $f$.

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

**Claim 2.10.** $D(KW_f) \leq depth(f)$

**Proof.** Let $F$ be a formula of depth $depth(f)$ 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 $f_0 \lor f_1$. Hence, $f_0(x) \lor f_1(x) = 1$ and $f_0(y) \lor f_1(y) = 0$. 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 $\land$, 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. \qed

**Claim 2.11.** $D(KW_f) \geq depth(f)$.

**Proof.** Let $P$ be the optimal protocol for $D(KW_f)$. We will convert its protocol tree into a formula as follows: each internal node in which Alice sends a bit is labelled by $\lor$ and each internal node in which Bob sends a bit is labelled by $\land$. 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) $\forall x \in S$,
Theorem 2.16. \[\text{DISJ} \to \Omega(n)\] and conclude from Theorem 2.14 that depth \(\Delta\to\) = \(\Omega(n)\) for any monotone function. Exercise: Show that \(\text{DISJ}_n\) can be computed by an OR-AND-OR monotone formula of size \(2^\Omega(n)\).

Theorem 2.16 (\cite{RW92}). \(\text{DISJ}_n\) \(\to \Omega(n)\), where \(n\) is the number of vertices.

Proof. We will show a randomized reduction from \(\text{DISJ}_n\) to \(\text{DISJ}_{n'}\) (for some \(n' = \Omega(n)\)) and conclude from Theorem 2.14 that depth \(\Delta\to\) = \(\Omega(n)\) = \(\Omega(n')\). In the above sequence, we use the result (to be shown later in the course) that \(\text{DISJ}_n\) = \(\Omega(n)\).
**Disjointness reduces to a covering problem:** Consider an instance of DisJ_n. Alice gets X ⊆ [n] and Bob gets Y ⊆ [n]. Alice makes the following graph on 3n vertices: V = {a_i, b_i, c_i : i ∈ [n]} and edges are added as follows, if i ∈ 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 /∈ Y then b_i ∈ T else c_i ∈ T (see Figure 1). Observe that X and Y are disjoint if and only if T covers the edges of Alice’s graph.

**Disjointness reduces to KW_{Match(n,3n)}:** The function Match(n, 3n) 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 KW_{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 ∈ T and considers T' = T \ {v}. Bob’s input for KW_{Match(n,3n)} is the complete bipartite graph between T' and {a_i, b_i, c_i : i ∈ [n]} \ 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 KW_{Match(n,3n)} on the input (G_A, G_B) and the protocol returns e = {x, y} ∈ E(G_A) \ E(G_B). If e is incident on σ(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 KW_{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.

**KW_{Match(n,3n)} reduces to KW_{Match_{4n}}:** Alice and Bob add n vertices (with identical names) to their graphs and connect them to all the other vertices.
Clearly, Alice’s graph has a perfect matching and Bob’s hasn’t. A protocol for $KW_{\text{Match}_n}$ must discover an edge $E(G_A) \setminus E(G_B)$, for all other (newly added) edges are the same in the two graphs.

References


Figure 3: Match($n, 3n$) to Match$_{4n}$