CS Theory (Spring '25) March 4, 2025

Lecture Note: Communication Lower Bounds

Instructor: *Josh Alman*

In class last time, we defined communication and we gave communication protocol for a few interesting problems, including the following:

Example 1. $\text{EQUALITY}(x,y) = \begin{cases} 1 & x = y \\ 0 & x \neq y \end{cases}$. *We showed an $O(n)$-communication protocol for* EQUALITY *last time.*

Example 2. $\text{MAJORITY}(x,y) = \begin{cases} 1 & \textit{if } xy \textit{ contains more 1's than 0's} \\ 0 & \textit{otherwise} \end{cases}$. *We showed an $O(\log n)$-communication protocol for* MAJORITY *last time.*

Example 3. $\text{PARITY}(x,y) = \begin{cases} 1 & \textit{if } xy \textit{ contains an odd number of } 1\textit{'s} \\ 0 & \textit{otherwise} \end{cases}$. *We showed an $O(1)$-communication protocol for* PARITY *last time.*

Our goal today is to give lower bounds, showing that these protocols are the best possible, in terms of amount of communication. We have showed last time we cannot do better than 2 bits of communication for PARITY, so our main goals are showing lower bounds for EQUALITY and PARITY.

The proof approach we will use is actually very similar to the the proof approach for streaming lower bounds. Recall that when we were proving streaming lower bounds, we gave two key definitions: distinguishable inputs and length-n distinguishing set (see the lecture note of February 25). The key argument was that if you have a length-n distinguishing set, then any streaming algorithm must assign different elements in it to different memory configurations. This means the size of the length-n distinguishing set is a lower bound on the number of possible memory configurations on length-n inputs, so the memory usage of the streaming algorithm is lower-bounded by the logarithm of the size of the distinguishing set.

We will give some analogous definitions for communication protocols: fooling pairs of inputs and length-n fooling set.

1 Communication Lower Bounds

We fix a function $f : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}$.

Definition 4 (Fooling pairs)**.** For strings $a, b, c, d \in \{0,1\}^*$, the pairs (a,b) and (c,d) are called *fooling pairs* for f if the two conditions hold:

- $f(a,b) = f(c,d)$.
- $f(a,d) \neq f(a,b)$ or $f(c,b) \neq f(a,b)$ (or both).

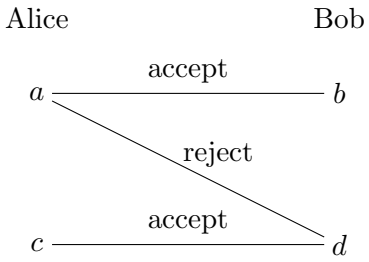

Figure 1: An illustration of fooling pairs

For example, one possible situation for fooling pairs (a, b) and (c, d) are illustrated in Figure 1. When the inputs to Alice and Bob are (a, b) or (c, d), they accept, but when the input is (a, d) they reject.

Definition 5. A *length-n fooling set* for f is a set of $S_n \subset \{0,1\}^* \times \{0,1\}^*$ such that the following two conditions hold:

- If $(a, b) \in S_n$, then $|a| \leq n/2$ and $|b| \leq n/2$.
- Any two distinct $(a, b), (c, d) \in S_n$ are fooling pairs for f.

We have the following theorem, which says that we will have a communication lower bound for input size n if we have a length-n fooling set. Note the similarity between this and the theorem we proved for streaming lower bounds (which says we have a streaming lower bound if we have a length-n distinguishable set).

Theorem 6. *If f has a length-n fooling set S_n, then any protocol P for f must use at least $(1/100) \log_2 |S_n|$ bits of communication for inputs of total length*[1] *at most n.*

To prove Theorem 6, we need the following definition of transcripts.

Definition 7 (Transcripts)**.** A *transcript* of a protocol P on input (a, b) is the entire message history of the protocol P running on input (a, b). Formally, it is of a sequence of pairs (sender, content), where each pair corresponds to a bit in the communication, the sender can be Alice or Bob, and the content can be 0 or 1.

The idea to prove Theorem 6 is to consider the map from input pairs to the transcript of the protocol P running on this input. If the number of transcripts on length-n input pairs is less than the size of a length-n fooling set, then by pigeonhole principle, there will be two fooling pairs that are mapped to the same transcript: but we will show this cannot happen. Therefore, the number of possible transcripts on length-n input pairs is at least the size of a length-n fooling set. We thus need to prove an upper bound on the number of transcripts based on the amount of communication, which is the following lemma.

Lemma 8. *For a protocol that uses k bits of communication, there are at most 4^k possible transcripts.*

Proof. For each of the k bits in the communication, it can be sent by Alice or Bob, and it can be either 0 or 1. Thus, for each bit, there are 4 possibilities. Therefore, for the k-bit communication, there are at most 4^k possible transcripts. □

[1] the sum of the lengths of Alice's and Bob's inputs

We have the next lemma to show that a fooling pair of inputs must correspond to different transcripts for a given protocol.

Lemma 9. *If (a, b) and (c, d) are a fooling pair of f, then for any protocol P for f, the transcripts of P running on (a, b) and (c, d) must be different.*

Proof. Since (a, b) and (c, d) are a fooling pair, we have $f(a, b) = f(c, d)$ and without loss of generality $f(a, d) \neq f(a, b)$. Assume to the contrary that P has the same transcript when running on (a, b) and (c, d).

Consider the transcript of P running on input (a, d). We will show that the transcripts are the same for the inputs (a, d), (a, b) and (c, d). We show this by induction.

Suppose the part of transcripts corresponding to the first k bits of communication are the same for the inputs (a, d), (a, b) and (c, d). Then, when Alice's input is a, Alice could not tell whether Bob's input is b or d within the first k bits of communication. Therefore, if the $(k + 1)$-th bit of communication is sent by Alice when the protocol runs on input (a, d), Alice will also send the $(k + 1)$-th bit when the input is (a, b), and the bit Alice sends will be the same no matter whether the input is (a, b) or (a, d). Similarly, when Bob's input is d, Bob could not tell whether Alice's input is a or c within the first k bits of communication, so if the $(k + 1)$-th bit of communication is sent by Bob when the protocol runs on input (a, d), Bob will also send the $(k + 1)$-th bit when the input is (c, d), and the bit Bob sends will be the same no matter whether the input is (a, d) or (c, d). To summarize, the part of the transcript corresponding to the first $k + 1$ bits are the same for the inputs (a, d), (a, b) and (c, d).

As the transcripts are the same for the inputs (a, d), (a, b) and (c, d), whether Alice accepts will be the same for the inputs (a, d) and (a, b), contradicting $f(a, d) \neq f(a, b)$. It follows that our assumption is wrong, so the transcript must be different for the inputs (a, b) and (c, d). □

Now we are ready to prove Theorem 6.

Proof of Theorem 6. If P uses less than $(1/100) \log_2 |S_n|$ bits of communication for every input of total length at most n, by Lemma 8, the total number of possible transcripts for those inputs is at most

$$\sum_{i=0}^{(1/100)\log_2 |S_n|} 4^i \leq 2 \cdot 4^{(1/100)\log_2 |S_n|} = 2 \cdot |S_n|^{1/50} < |S_n|.$$

Thus, by pigeonhole principle, there are two pairs in S_n that have the same transcript under the protocol P. This contradicts Lemma 9 □

Below we use Theorem 6 to prove communication lower bounds for EQUALITY and MAJORITY.

Theorem 10. *Any protocol for* EQUALITY *must use at least $n/1000$ bits of communication.*

Proof. Let $S_n \coloneqq \{(a, a) \mid a \in \{0, 1\}^* \text{ and } |a| = \lfloor n/2 \rfloor\}$. This is a length-$n$ fooling set, as for any two pairs (a, a), (b, b) in S_n, $\text{EQUALITY}(a, a) = \text{EQUALITY}(b, b) = 1$ but $\text{EQUALITY}(a, b) = 0$. Therefore, by Theorem 6, any protocol for EQUALITY needs at least

$$\frac{1}{100} \log_2 |S_n| = \frac{1}{100} \log_2 (2^{\lfloor n/2 \rfloor}) \geq \frac{n}{1000}$$

bits of communication. □

Theorem 11. *Any protocol for* MAJORITY *must use at least* $(1/1000)\log_2 n$ *bits of communication.*

Proof. Let $m \coloneqq \lfloor n/2 \rfloor$, and let $S_n \coloneqq \{(a,b) \mid a = 1^k 0^{m-k}, b = 1^{m-k+1}0^{k-1}, 1 \le k \le m\}$. This is a length-$n$ fooling set, as for any two pairs $(1^k 0^{m-k}, 1^{m-k+1}0^{k-1})$, $(1^\ell 0^{m-\ell}, 1^{m-\ell+1}0^{\ell-1})$ in S_n, supposing $k < \ell$ without loss of generality, we have $\text{MAJORITY}(1^k 0^{m-k}, 1^{m-k+1}0^{k-1}) = \text{MAJORITY}(1^\ell 0^{m-\ell}, 1^{m-\ell+1}0^{\ell-1}) = 1$ but $\text{MAJORITY}(1^k 0^{m-k}, 1^{m-\ell+1}0^{\ell-1}) = 0$. Therefore, by Theorem 6, any protocol for MAJORITY needs at least

$$\frac{1}{100}\log_2 |S_n| = \frac{1}{100}\log_2 m \ge \frac{1}{1000}\log_2 n$$

bits of communication. □