Electronic Colloquium on Computational Complexity, Report No. 2 (2017)

Some notes on two lower bound methods for communication complexity

Frantisek Duris*

Faculty of Mathematics, Physics and Informatics,
Comenius University,
Mlynska dolina 842 48 Bratislava

December 8, 2016

Abstract

We compare two methods for proving lower bounds on standard two-party model of communication complexity, the Rank method and Fooling set method. We present bounds on the number of functions $f(x, y)$, $x, y \in \{0, 1\}^n$, with rank of size k and fooling set of size at least k, $k \in [1, 2^n]$. Using these bounds we give a novel proof that almost all Boolean functions f are hard, i.e., the communication complexity of f is greater than or equal to n, using the field $\mathbb{Z}_2$.

Keywords: communication complexity, Rank method, Fooling set method

*fduris@dcs.fmph.uniba.sk

ISSN 1433-8092

1 Introduction and layout of the paper

Communication complexity studies the amount of communication bits exchanged between two or more parties in order to compute some given Boolean function. The simple two-party model was first defined by Yao (1979) in which the two parties aim to evaluate a Boolean function $f(x, y)$, where x is the input of one party and y is the input of the second party (in the two-party setting, the parties are usually referred to as Alice and Bob). The communication complexity measures the minimal number of bits needed to be exchanged between the parties in order to compute the function f. While the model looks very minimalistic, it captures many fundamental issues related to lower bounds on the complexity of communication as well as complexity measures for other computational models, e.g., finite automata, Turing, machines, VLSI and Boolean circuits, etc. (for other examples see, for example, Kushilevitz, 1997). Furthermore, several extensions of the standard model were proposed. In randomized communication complexity, parties are allowed to toss a coin to decide which messages they send, in the variable partition model the parties are allowed to freely partition the whole input among them, in non-deterministic model the parties can choose the messages they send, and we only require that there is an execution of protocol that leads to the correct answer.

In this paper we consider the standard deterministic two-party model, and we analyse two of the basic lower-bound techniques, the Fooling set method due Yao (1979); Lipton and Sedgewick (1981) and the Rank method due Mehlhorn and Schmidt (1982). Other methods include Tiling method due Yao (1981), or Rectangle size method due Karchmer et al. (1992). The proper understanding of the lower-bound techniques is essential, because for a given function f not every method gives equal or optimal bounds. For example, if we let $cc(f)$, $t(f)$, $r(f)$ and $fs(f)$ stand for communication complexity of a function f, and its lower bounds by Tiling, Rank, and Fooling set method, respectively, then it was proved by Aho et al. (1983) that for every Boolean function f

- $t(f) - 1 \leq cc(f) \leq (t(f) - 1)^2$,
- $r(f) \leq t(f)$ and $fs(f) \leq t(f)$.

Furthermore, Dietzfelbinger et al. (1996) proved that $fs(f) \leq (r(f) + 1)^2$ for all Boolean functions. Friesen and Theis (2012) proved that this bound is asymptotically tight for fields of non-zero characteristic, and Hamed and Lee (2013) proved the same result for fields of zero characteristic. Dietzfelbinger et al. (1996) further showed that there exists Boolean functions $f_1(x, y)$ and $f_2(x, y)$, $x, y \in \{0, 1\}^n$, such that

- $cc(f_1) = r(f_1) = n \geq log_2(10n) \geq fs(f_1)$,
- $\frac{log_2 3}{2} n = r(f_2) < fs(f_2) = cc(f_2) = n$.

In this paper we present bounds on the number of functions $f(x, y)$, $x, y \in \{0, 1\}^n$, with $r(f) = k$ and $fs(f) \geq k$ for some $k \in [1, 2^n]$ (*Theorems 3.2* and *4.3*). Using these bounds we give novel proof that almost all Boolean functions $f(x, y)$ are hard, i.e., $cc(f) \geq n$, using the field $\mathbb{Z}_2$ (*Corollary 3.3*). The Rank method which we use in this proof works over any field, and currently the proof that almost all functions are hard is based on field $\mathbb{Q}$ (*Remark 3.4*).

2 Elementary definitions and theorems

Let f be a two-argument boolean function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$. Let Alice and Bob be the two communicating parties. Alice is given an input $x \in \{0,1\}^n$ and Bob is given the input $y \in \{0,1\}^n$. They wish to compute the value of $f(x,y)$. The computation is done using a communication protocol which specifies who sends a message to whom and what is the content of the message. A communication protocol P computes the function f, if for every input pair $(x,y) \in \{0,1\}^n \times \{0,1\}^n$ the protocol terminates with the value $f(x,y)$ as its output. Let $s_P(x,y) = \{m_1, m_2, ..., m_r\}$ be the communication exchanged on input (x,y) during the execution of P, where m_i denotes the i^{th} message sent in the protocol. Let m_i denote the length of m_i in bits, and let $|s_P(x,y)| = \sum_{i=1}^{r} |m_i|$. The deterministic communication complexity of protocol P is defined as

$$D(P) = \max_{(x,y)\in\{0,1\}^n\times\{0,1\}^n} |s_P(x,y)|.$$

The deterministic communication complexity of a function f is defined as

$$D(f) = \min_{P:P \text{ computes } f} D(P).$$

Trivially, all functions can be computed by a protocol which transfers the whole input of Alice to Bob who can now compute $f(x,y)$ and send the result back to Alice. Thus, $D(f) \leq n+1$. A function f is called hard, if $D(f) \geq n$.

Definition 2.1 (Fooling set, Yao, 1979; Lipton and Sedgewick, 1981)**.** A set of input pairs $\{(x_1,y_1),(x_2,y_2),...,(x_l,y_l)\}$ is called a fooling set (of size l) with respect to f, if there exists $b \in \{0,1\}$ such that

1. For all i, $f(x_i,y_i) = b$.
2. For all $i \neq j$, either $f(x_i,y_j) \neq b$ or $f(x_j,y_i) \neq b$.

Note that we will call the input pairs (x_i,y_j) and (x_j,y_i) symmetric pairs to (x_i,y_i) and (x_j,y_j) from the fooling set.

Lemma 2.2 (Fooling set method, Yao, 1979; Lipton and Sedgewick, 1981)**.** *Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ be a function. If there exists a fooling set of size l with respect to f, then*

$$D(f) \geq \log_2 l.$$

Lemma 2.3 (Rank method, Mehlhorn and Schmidt (1982))**.** *Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ be a function. Let M_f be a matrix representation of f, where each row and column is associated with input $x \in \{0,1\}^n$ and $y \in \{0,1\}^n$, respectively, and the (x,y) entry in M_f holds the value $f(x,y)$. Then,*

$$D(f) \geq \log_2 \left(2 \cdot rank(M_f) - 1\right).$$

3 Rank method

In this section we describe the number of Boolean functions $f(x, y), x, y \in \{0, 1\}^n$, with communication complexity at least some k that can be exposed by the Rank method (i.e., functions f such that $rank(M_f) = k$). We derive an exact number (*Theorem 3.1*) as well as asymptotically tight and more practical bounds (*Theorem 3.2*). From these bounds we derive that for sufficiently large n almost all functions f are hard (*Corollary 3.3*).

Theorem 3.1. *The number of matrices of size $2^n \times 2^n$ over the field $\mathbb{Z}_2$ with rank k is*

$$R(k) = \left(\prod_{i=1}^{k} \left(2^{2^n} - 2^{i-1}\right)\right) \sum_{\substack{(s_0, s_1, \ldots, s_k) \\ s_0+s_1+\ldots+s_k = 2^n - k}} \prod_{j=0}^{k} \left(2^j\right)^{s_j}. \tag{3.1}$$

Proof. We prove this equation by showing how to construct each matrix of rank k once and only once. By counting the choices during this construction we get the the number of matrices. The construction consists of two steps. First, we need to select k linearly independent binary vectors of size 2^n to put in k rows. Note that we do not yet care about their positions in the matrix, but we do care about their order – they will appear in the matrix in this order. One can easily check that there are

$$\prod_{i=1}^{k} \left(2^{2^n} - \sum_{j=0}^{i-1} \binom{i-1}{j}\right) = \prod_{i=1}^{k} \left(2^{2^n} - 2^{i-1}\right)$$

possibilities – once we selected i^{th} vector, we remove all its linear combinations with the already selected vectors from the possible choices for the remaining $k - i$ rows. Second, we need to select the positions of these linearly independent vectors in the matrix and also what will be in the matrix between them. The k linearly independent rows create altogether at most $k + 1$ gaps between each of them or above/below the first/last row. Let $\alpha_{i_1}, \alpha_{i_2}, ..., \alpha_{i_k}$ be linearly independent vectors where i_j is number of the row in which the vector α_{i_j} lies. Then, schematically, we have for the constructed matrix M

$$M = \begin{pmatrix} \text{gap no. } 0 \\ \alpha_{i_1} \\ \text{gap no. } 1 \\ \alpha_{i_2} \\ \cdots \\ \text{gap no. } k-1 \\ \alpha_{i_k} \\ \text{gap no. } k \end{pmatrix}.$$

The number of all rows that lie in these gaps is clearly $2^n - k$. We fill the l^{th} gap with linear combinations of vectors $\alpha_{i_1}, ..., \alpha_{i_l}$ (i.e., the set of l linearly independent vectors that lie above the l^{th} gap). Note that the *gap no. 0*, if it exists, must contain only zero-only rows. It follows that for the l^{th} gap we have $\sum_{i=0}^{l}\binom{l}{i} = 2^l$ different vectors that it can contain. Furthermore, if the size of the l^{th} gap is s_l, then we have $\left(2^l\right)^{s_l}$ possibilities for this gap. Thus, for all gaps we have

$$\sum_{\substack{(s_0,s_1,\ldots,s_k)\\ s_0+s_1+\ldots+s_k=2^n-k}} \prod_{j=0}^{k} \left(2^j\right)^{s_j}$$

possibilities. Note that the positions of the linearly independent vectors in the matrix (not yet explicitly assigned) are determined by the sizes of the gaps. Finally, we show that we can construct each matrix of rank k once and only once.

Completeness. Let M be any matrix with rank k. Let Λ be an empty set. Go through each row of M (from top to bottom), and if the current row is

1. a linear combination of vectors in the set Λ or zero-only vector, then do nothing;

2. a vector linearly independent with the vectors in the set Λ, then add this vector to Λ.

After this procedure is finished, the set Λ contains k linearly independent vectors (any other number would contradict the rank of M). Moreover, every row either lies in Λ or is a linear combination of rows that lie higher in the matrix M and are in Λ. Specifically, vectors that lie above vector α_{i_1} are zero-only vectors. Hence, every matrix M can be constructed in our way.

Uniqueness. This also follows from the construction above because if we could construct some matrix in two different ways, then they would have to agree at least in the number of zero-only rows on the top of the matrix M. But then, they would also have to agree on the first linearly independent vector and in the index of its row. Since the procedure is deterministic, that is no way in which the computation would split in two separate ways. □

Theorem 3.2. *Let $R(k)$ be the number of matrices of size $2^n \times 2^n$ over the field $\mathbb{Z}_2$ with rank k from Theorem 3.1. Then, it holds*

$$0.288788 \cdot 2^{2k2^n-k^2} \leq R(k) \leq 2 \cdot 2^{2k2^n-k^2}.$$

Proof. Consider the first part of the equation (3.1). We have

$$2^{k2^n} \prod_{i=1}^{2^n}\left(1-\frac{1}{2^i}\right) \leq \prod_{i=1}^{k}\left(2^{2^n}-2^{i-1}\right) \leq 2^{k2^n}$$

Furthermore, if the series $\sum_{i=1}^{\infty} a_i$ converges, then so does the product $\prod_{i=1}^{\infty}(1-a_i)$. Thus, $\lim_{n\to\infty}\prod_{i=1}^{2^n}\left(1-\frac{1}{2^i}\right)$ exists, and it can be numerically determined to be 0.288788... . The second part of the equation (3.1) we can rewrite as

$$\sum_{\substack{(s_0,s_1,\ldots,s_k)\\ s_0+s_1+\ldots+s_k=2^n-k}} \prod_{j=0}^{k}\left(2^j\right)^{s_j} = \sum_{\substack{(s_0,s_1,\ldots,s_k)\\ s_0+s_1+\ldots+s_k=2^n-k}} 2^{\lg\prod_{j=0}^{k}2^{js_j}} = \sum_{\substack{(s_0,s_1,\ldots,s_k)\\ s_0+s_1+\ldots+s_k=2^n-k}} 2^{\sum_{j=0}^{k} js_j}.$$

Since $\sum_{i=0}^{n} 2^i = 2^{n+1} - 1$, we can easily bound this expression as follows

$$2^{(2^n-k)k} \leq \sum_{\substack{(s_0,s_1,\ldots,s_k) \\ s_0+s_1+\ldots+s_k=2^n-k}} 2^{\sum_{j=0}^{k} j s_j} \leq 2 \cdot 2^{(2^n-k)k},$$

because the maximum value of the sum $\sum_{j=0}^{k} j s_j$ is $(2^n - k)\,k$. The lemma follows. □

Corollary 3.3. *A randomly chosen Boolean function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ is hard with probability tending to 1 as n approaches infinity.*

Proof. Let M_f be a matrix representation of function f as in *Lemma 2.3*. For a function f to be hard, it suffices that $rank(M_f) \geq 2^{n-2} + 1$, because

$$D(f) \geq \log_2 (2 \cdot rank(M_f) - 1) \geq \log_2 \left(2^{n-1} + 1\right) = (n-1) + \log_2 \left(1 + \frac{1}{2^{n-1}}\right) \geq n,$$

since $D(f)$ has to be a natural number. Let $R(k)$ be from *Theorem 3.2*. The number of matrices with $rank(M_f) \geq 2^{n-1} + 1$ is lower bounded by $2^{2^{2n}} - \sum_{i=1}^{2^{n-2}} R(i)$ which, when compared with the number of all functions $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, yields

$$\lim_{n\to\infty} \frac{2^{2^{2n}} - \sum_{i=1}^{2^{n-2}} R(i)}{2^{2^{2n}}} \geq \lim_{n\to\infty} \frac{2^{2^{2n}} - 2^{n-2} R(2^{n-2})}{2^{2^{2n}}} = \lim_{n\to\infty} \frac{2^{2^{2n}} - 2^{n-2} 2^{2^{n-1} * 2^n - 2^{2(n-2)} + 1}}{2^{2^{2n}}} =$$

$$= \lim_{n\to\infty} \frac{2^{2^{2n}} - 2^{\frac{7}{16} 2^{2n} + n - 1}}{2^{2^{2n}}} = 1.$$

□

Remark 3.4. Komlós (1967, 1968) proved that a random (0,1) $n \times n$ matrix has rank n over the field $\mathbb{Q}$ with probability tending to 1 for $n \to \infty$. This means, that almost all functions are hard and the Rank method almost always gives the optimal lower bound as was concluded in Dietzfelbinger et al. (1996). The *Corollary 3.3* proves the same result using only the field $\mathbb{Z}_2$.

4 Fooling set method

In this section we describe the number of Boolean functions $f(x,y), x, y \in \{0,1\}^n$, with communication complexity at least some k that can be exposed by the Fooling set method (i.e., functions f such that $fs(f) \geq k$). The upper bound as well as conclusion that almost no functions f has large fooling sets appeared in Dietzfelbinger et al. (1996). Here we prove the lower bound.

Theorem 4.1 (Dietzfelbinger et al., 1996)**.** *Let $F(k)$ be the number of matrices of size $2^n \times 2^n$ over the field $\mathbb{Z}_2$ with fooling set of size at least k. Then, it holds*

$$F(k) \leq 2 \cdot \binom{2^n}{k}^2 \cdot k! \cdot 3^{\binom{k}{2}} \cdot 2^{2^{2n} - k^2}.$$

Remark 4.2. Note that at least for $k \leq \frac{n-2}{1.2075}$ this bound is weaker than $2^{2^{2n}}$, i.e., number of all matrices of size $2^n \times 2^n$. By using the simple lower bounds we get

$$2 \cdot \binom{2^n}{k}^2 \cdot k! \cdot 3^{\binom{k}{2}} \cdot 2^{-k^2} \geq 2 \cdot \left(\frac{2^n}{k}\right)^{2k} \cdot e \left(\frac{k}{e}\right)^k \cdot 2^{\frac{1}{2}(k^2-k)\lg 3 - k^2} =$$

$$= 2^{1+2kn+\lg e+\frac{1}{2}(k^2-k)\lg 3-k^2-k\lg k-k\lg e}.$$

If this expression is greater than 1, then the upper bound from *Theorem 4.1* is weak. We can rewrite this into an inequality $\left(-1+\frac{1}{2}\lg 3\right)k^2 - k\lg k + \left(2n - \frac{1}{2}\lg 3 - \lg e\right)k + 1 + \lg e \geq 0$ and then $\left(-1+\frac{1}{2}\lg 3\right)k^2 + \left(2n - \frac{1}{2}\lg 3 - \lg e\right)k + 1 + \lg e \geq k\lg k$. By replacing the right-hand side with k^2 (which would still imply that the left-hand side is greater or equal to $k\lg k$), we get a simple quadratic inequality

$$\left(-2+\frac{1}{2}\lg 3\right)k^2 + \left(2n - \frac{1}{2}\lg 3 - \lg e\right)k + 1 + \lg e \geq 0.$$

Let $a = \left(-2+\frac{1}{2}\lg 3\right)$, $b = \left(2n - \frac{1}{2}\lg 3 - \lg e\right)$, and $c = (1+\lg e)$. Then, because $a < 0$, the solution interval of this inequality (i.e., for which k the upper bound from *Theorem 4.1* is weak) is

$$k \in \left(0,\ \left(-b-\sqrt{b^2-4ac}\right)/2a\right],$$

because $\left(-b+\sqrt{b^2-4ac}\right)/2a < 0$, and, of course, k must be positive. Finally, with $a' = -a$ we have for the right end of the solution interval

$$\frac{-b-\sqrt{b^2-4ac}}{2a} = \frac{b+\sqrt{b^2+4a'c}}{2a'} = b \cdot \left[\frac{1+\sqrt{1+\frac{4a'c}{b^2}}}{2a'}\right] \geq \frac{b}{a'} \geq \frac{n-2}{1.2075}.$$

Theorem 4.3. *Let $F(k)$ be the number of matrices of size $2^n \times 2^n$ over the field $\mathbb{Z}_2$ with fooling set of size at least k. Then, it holds*

$$F(k) \geq 2 \cdot k! \cdot 2.5^{\binom{k}{2}} \cdot 2^{2^{2n}-k^2}.$$

Proof. The upper bound for $F(k)$ from *Theorem 4.1* includes many repetitions due to terms $\binom{2^n}{k}^2$ and $2^{2^{2n}-k^2}$ (i.e., we count many matrices more than once because they have multiple fooling sets of size k). In order to eliminate these type of repetitive counting, we restrict ourselves to count only the matrices that have only one fooling set of size k in the top-left corner of size $k \times k$, thus omitting the term $\binom{2^n}{k}^2$. Schematically,

$$M = \begin{pmatrix} A_{k\times k} & B \\ C & D \end{pmatrix}$$

where the number of fooling sets of size k in A is 1. Thus, we only need to ensure that we avoid all matrices $k \times k$ with two (or more) fooling sets of size k. We will call this top left submatrix of M as A_k. Observe that if only the elements from the fooling set had the value b in the matrix A_k, and the rest of the elements had value $1-b$, then each different fooling set would correspond with a different matrix A_k. Thus, the repetitions are due to the term $3^{\binom{2^n}{k}}$ – the choices for the symmetric pairs (see *Definition 2.1*). Previously, we allowed all of the following possibilities $\{(1-b, 1-b), (b, 1-b), (1-b, b)\}$ for any pair of symmetric elements. We argue that if we restrict our choices to just $\{(1-b, 1-b), (b, 1-b)\}$, then the repetitions will vanish. Note that the notation $(b, 1-b)$ means that the top (with respect to matrix A_k) element of the symmetric pair is b and the bottom is $1-b$. Consider, on the contrary, that there is matrix A_k that even with the limited choices, it has two fooling sets Λ and Δ of size k. Assume that Λ and Δ differ in the choice for the last row of m_k. Let the configuration of these fooling sets in A_k be

$$A_k = \begin{pmatrix} \cdots & \cdots & \cdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & \cdots & \cdots & b^l_\lambda & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & b^k_\lambda & \cdots & b_\delta & \cdots \end{pmatrix}.$$

In this configuration, b_δ is a bottom symmetric element of the pair $(b^l_\lambda, b^k_\lambda)$ from the fooling set Λ. But this contradicts the allowed choices for Λ, because the bottom symmetric element has value b (the configuration in which b_δ is before b_λ in the last row follows analogously). Thus, Λ and Δ must agree on the element from the last row, and by the induction argument they must agree everywhere. From this it follows that

$$F(k) \geq 2 \cdot k! \cdot 2^{\binom{k}{2}} \cdot 2^{2^{2n}-k^2}.$$

To improve the lower bound, consider a submatrix 3×3 with six elements b and three elements $1-b$. One can easily check that in each such submatrix, there are two different fooling sets of size three as in the this example below

$$\begin{pmatrix} b_\lambda & b_\delta & 1-b \\ 1-b & b_\lambda & b_\delta \\ b_\delta & 1-b & b_\lambda \end{pmatrix}.$$

Note that there are twelve such submatrices altogether, or six if we do not distinguish between the fooling sets. Furthermore, one can easily check that in each such 3×3 submatrix at least one previously unallowed symmetric pair was used (i.e., value b in the bottom corner). Therefore, each unallowed pair can possibly double the number of fooling sets in the matrix A_k. On the other hand, if there are x unallowed pairs, then this cannot create more than 2^x of additional fooling sets. The x unallowed symmetric pairs provide x additional values b, and each additional fooling set has to have a different choice of these values b (because allowed symmetric pairs do not create multiple fooling sets), which altogether is at most 2^x possibilities. Thus, we get a better lower bound on the number of matrices with fooling set of size k in the top-left corner A_k by allowing all three choices for symmetric pairs but penalising each usage of the previously unallowed symmetric pair by a factor $\frac{1}{2}$. Formally, we have this expressions from the upper bound (*Theorem 4.1*) by omitting the term $\binom{2^n}{k}^2$ (fooling set is in the top-left corner)

$$2 \cdot k! \cdot 3^{\binom{k}{2}} \cdot 2^{2^{2n}-k^2} = 2 \cdot k! \cdot \sum_{i=0}^{\binom{k}{2}} \binom{\binom{k}{2}}{i} 2^{\binom{k}{2}-i} \cdot 2^{2^{2n}-k^2},$$

where i determines how many times we used the previously unallowed symmetric pair, and with the penalising factor $\frac{1}{2^i}$we get

$$F(k) \geq 2 \cdot k! \cdot \sum_{i=0}^{\binom{k}{2}} \binom{\binom{k}{2}}{i} 2^{\binom{k}{2}-i} \frac{1}{2^i} \cdot 2^{2^{2n}-k^2} = 2 \cdot k! \cdot 2.5^{\binom{k}{2}} \cdot 2^{2^{2n}-k^2}.$$

□

References

A. V. Aho, J. D. Ullman, and M. Yannakakis. On notions of information transfer in vlsi circuits. In *Proceedings of the fifteenth annual ACM symposium on Theory of computing*, pages 133–139. ACM, 1983.

M. Dietzfelbinger, J. Hromkovič, and G. Schnitger. A comparison of two lower-bound methods for communication complexity. *Theoretical Computer Science*, 168(1):39–51, 1996.

M. Friesen and D. O. Theis. Fooling sets (aka cross-free matchings) and rank in non-zero characteristic. *arXiv preprint arXiv:1208.2920*, 2012.

A. Hamed and T. Lee. Rank and fooling set size. *arXiv preprint arXiv:1310.7321*, 2013.

M. Karchmer, E. Kushilevitz, and N. Nisan. Fractional covers and communication complexity. In *Structure in Complexity Theory Conference, 1992., Proceedings of the Seventh Annual*, pages 262–274. IEEE, 1992.

J. Komlós. On the determinant of/0, 1/ matrices(number of completely nonsingular n x n matrices with elements 0, 1). *Studia scientiarum mathematicarum Hungarica*, 2(1):7–21, 1967.

J. Komlós. On the determinant of random matrices. *Studia Sci. Math. Hungar*, 3(4):387–399, 1968.

E. Kushilevitz. Communication complexity. *Advances in Computers*, 44:331–360, 1997.

R. J. Lipton and R. Sedgewick. Lower bounds for vlsi. In *Proceedings of the thirteenth annual ACM symposium on Theory of computing*, pages 300–307. ACM, 1981.

K. Mehlhorn and E. M. Schmidt. Las vegas is better than determinism in vlsi and distributed computing. In *Proceedings of the fourteenth annual ACM symposium on Theory of computing*, pages 330–337. ACM, 1982.

A. C. Yao. The entropic limitations on vlsi computations. In *Proceedings of the thirteenth annual ACM symposium on Theory of computing*, pages 308–311. ACM, 1981.

A. C.-C. Yao. Some complexity questions related to distributive computing (preliminary report). In *Proceedings of the eleventh annual ACM symposium on Theory of computing*, pages 209–213. ACM, 1979.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il