CUCS-73-83

LOWER BOUNDS ON COMMUNICATION COMPLEXITY

Pavol Duris*
Slovak Academy of Science

Zvi Galil*
Department of Computer Science
Columbia University
Tel-Aviv University

Georg Schnitger
Department of Computer Science
Penn State University

October 1983

*Research supported by National Science Foundation Grant MCS-8303139.

Abstract: We prove the following four results on communication complexity:

1) For every $k \geq 2$, the language L_k of encodings of directed graphs of out degree one that contain a path of length $k + 1$ from the first vertex to the last vertex and can be recognized by exchanging $O(k \log n)$ bits using a simple k-round protocol requires exchanging $\Omega(n^{1/2}/k^4 \log^3 n)$ bits if any (k-1)-round protocol is used.

2) For every $k \geq 1$ and for infinitely many $n \geq 1$, there exists a collection of sets $L_k^n \subseteq \{0,1\}^{2n}$ that can be recognized by exchanging $O(k \log n)$** bits using a k-round protocol, and any (k-1)-round protocol recognizing L_k^n requires exchanging $\Omega(n/k)$ bits.

3) Given a set $L \subseteq \{0,1\}^{2n}$, there is a set $\tilde{L} \subseteq \{0,1\}^{8n}$ such that any (k-round) protocol recognizing $\tilde{L}$ can be transformed to a (k-round) fixed partition protocol recognizing L with the same communication complexity, and vice versa.

4) For every integer function f, $1 \leq f(n) \leq n$, there are languages recognized by a one round deterministic protocol exchanging $f(n)$ bits, but not by any nondeterministic protocol exchanging $f(n) - 1$ bits.

The first two results show in an incomparable way an

*All logarithms in the paper are of base 2.

exponential gap between (k-1)-round and k-round protocols, settling a conjecture by Papadimitriou and Sipser. The third result shows that as long as we are interested in existence proofs, a fixed partition of the input is not a restriction. The fourth result extends a result by Papadimitriou and Sipser who showed that for every integer function f, $1 \leq f(n) \leq n$, there is a language accepted by a deterministic protocol exchanging $f(n)$ bits but not by any deterministic protocol exchanging $f(n) - 1$ bits.

0. Introduction

Suppose that a language $L \subseteq \{0,1\}^*$ must be recognized by two distant computers. Each computer receives half of the input bits, and the computation proceeds using some protocol for communication between the two computers. The minimum number of bits that has to be exchanged in order to successfully recognize $L \cap \{0,1\}^{2n}$, minimized over all partitions of the input bits into two equal parts, and considered as a function of n, is called the communication complexity of L.

This model was suggested by Papadimitriou and Sipser [1]. They motivated it by pointing out its relation to lower bound proofs in VLSI [2,3,4]. A closely related model, where the partition of the input is fixed was studied in [5,6]. Both versions were also studied in [7,8].

We now review the model [1]:

A protocol on 2n inputs is a pair $D_n = (\pi,\phi)$, where

(a) π is a partition of $\{1,2,\ldots,2n\}$ into two equal sets S_I and S_{II} (this corresponds to the partition of the input into the two halves for the two computers); and

(b) ϕ is a function from $\{0,1\}^n \times \{0,1,\$\}^*$ to $\{0,1\}^* \cup \{\text{accept, reject}\}$. Intuitively, the first argument of ϕ is the local part of the input, while the second argument is the "log" of all previous messages, with \$ serving as the delimiter between messages. The result of ϕ is the next message. For a given string $c \in \{0,1,\$\}^*$, the function ϕ has the property that for every two $y,y' \in \{0,1\}^n$, $\phi(y,c)$ is not a proper prefix of $\phi(y',c)$ and if $\phi(y,c) \in \{\text{accept}, \text{reject}\}$ then $\phi(y',c) = \phi(y,c)$. (This second part is mistakenly missing in [1].) This *prefix-freeness* property assures that the exchanged messages are *self-delimiting*, and that no extra "end of transmission" symbol is required.

A *computation of* D_n *on input* $x \in \{0,1\}^{2n}$ is a string $c = c_1\$c_2\$\cdots\$c_k\$c_{k+1}$, where $k \geq 0$, $c_1,\ldots,c_k \in \{0,1\}^*$, $c_{k+1} \in \{\text{accept, reject}\}$, and such that, for each integer ℓ, $0 \leq \ell < k$, we have: (1) if ℓ is odd, then $c_{\ell+1} = \phi(x_I, c_1\$c_2\$\cdots\$c_\ell)$, where x_I is the input x restricted to the set S_I; and (2) if ℓ is even, then $c_{\ell+1} = \phi(x_{II}, c_1\$c_2\$,\ldots,\$c_\ell)$.

In other words, in a computation the two computers take turns computing the next message to be sent, by consulting the local input and all previous exchanges (and using, without loss of generality,the same function ϕ). Obviously, this process is completely deterministic. The *length* of a computation c is the total length of all messages in c (ignoring \$'s and the final accept/reject).

Let $L \subseteq \{0,1\}^{2n}$ be a language, and D_n be a deterministic protocol. We say D_n *recognizes* L if, for each $x \in \{0,1\}^{2n}$, the computation of D_n on input x is always finite, and ends with accept iff $x \in L$. Let f be a function from integers to integers. We say that L is recognizable within communication f, $L \in \mathrm{COMM}(f(n))$, if there is a protocol D_n recognizing L such that for all $x \in \{0,1\}^{2n}$ the computation of D_n on x has length at most $f(n)$.

Let $L \subseteq \{0,1\}^*$ be a language, $\Delta = \langle D_n \rangle$ a sequence of deterministic protocols and f a function from integers to integers. We say that Δ recognizes L (L is recognizable within communication f, $L \in \mathrm{COMM}(f(n))$) if D_n recognizes $L^n \equiv L \cap \{0,1\}^{2n}$ (if $L^n \in \mathrm{COMM}(f(n))$) for all n.

The prefixfreeness property is motivated in [1]. We need it only for our last result where we want to pin down exactly the communication complexity. In other cases we augment the messages with an endmarker. We do not change the definition of the length of the message. Even if we counted it in the first three results we would at most double the communication complexity.

We also consider nondeterministic protocols and the corresponding class NCOMM(f(n)). In nondeterministic protocols φ is a "nondeterministic function"; i.e. it may have several values (and therefore it is not a function). The definitions above apply if whenever we write $\varphi(x,c)$ we mean a possible value of $\varphi(x,c)$.

In [1], Papadimitriou and Sipser gave two open problems. The first is related to their main result in which they showed a language $L \in NCOMM(\log n) - COMM(cn)$, for some $c > 0$; i.e. an exponential gap between deterministic and nondeterministic protocols. However, $\bar{L} \notin NCOMM(cn)$ and they asked whether there is a language in $NCOMM(\log n) \cap co\text{-}NCOMM(\log n) - COMM(cn)$ (i.e., whether a language such that it and its complement are easy nondeterministically but exponentially harder deterministically). Recently, Aho Ullman and Yannakakis [8] answered the question affirmatively.

Papadimitriou and Sipser defined the notion of k-round protocols in which up to k messages are exchanged. They denoted $COMM_k(f(n))$ and $NCOMM_k(f(n))$ the corresponding classes of languages when we restrict ourselves to k-round protocols. They defined the languages $L_k = \{w_0 w_1 \ldots w_{2^m-1} | w_i \in \{0,1\}^m$ and $\exists j_0, \ldots, j_{k+1} | w_{j_i} = j_{i+1}$ where $j_0 = 0$ and $j_{k+1} = 2^m-1\}$. A member of L_k encodes a directed graph of outdegree 1 having a path of length k + 1 from vertex 0 to vertex 2^m-1. It is easily seen that $L_2 \in COMM_2(2 \log n)$ and in fact $L_k \in COMM_k(k \log n)$. They showed that $L_2 \notin COMM_1(\sqrt{n}/(2 \log n))$, thus exhibiting an exponential gap between one- and two-round protocols. The second open problem in their paper was whether a similar gap exists between k- and (k-1)-round protocols. They conjectured that indeed this is the case and that L_k is the witness to this fact.

Conjecture [1]: For every $k > 1$, there is an ℓ such that

$$L_k \notin \mathrm{COMM}_{k-1}(n^{1/\ell}).$$

Our first two results show that indeed there is an exponential gap between k- and (k-1)-round protocols. Theorem 1, settles the conjecture above in almost the strongest sense.

Theorem 1: For every $k \geq 1$, $L_{k+1} \notin \mathrm{COMM}_k(n^{1/2}/(36k^4 \log^3 n))$.

Remark: One can easily show that $L_{k+1} \in \mathrm{COMM}_k(n^{1/2}/(\log n)^{1/2})$.

The proof of Theorem 1 is combinatorial. We have found a way to "force" our intuition. For many open problems in computational complexity the solution is intuitively clear: Intuitively, $P \neq NP$ because we must check all assignments when solving SATISFIABILITY. Unfortunately, we can rarely transform such an intuition into a proof.

In our case, our intuition tells us that if the two computers have the wrong vertices, they must exchange $k + 1$ internal vertices in order to check whether a path of length $k + 2$ exists. So, if only k rounds are allowed, the computer that is supposed to make the decision will be "one vertex behind". The other computer will have to send him a long list of values not knowing what is the (k+1)-st vertex on the path.

Of course, our computers do not necessarily get the wrong vertices neither do they always exchange vertices. What is worse, the input is partitioned arbitrarily and each computer may get only part of the bits of the various edges. We found a way around it. By restricting attention to a subset of the inputs which is large enough we were able to find graphs with k + 2 layers such that, indeed, the two computers have the wrong vertices. Starting with this subset of inputs we fix a certain path by adding one vertex at a time. Each time we further restrict the inputs to contain this initial path, say of length i, and to have the same i messages exchanged. Not allowing long enough messages, we are still left with a large number of such inputs, so after k messages the remaining set has both: inputs in L_{k+1} and not in L_{k+1}, because some initial paths have the completing edge and some don't. All of this is achieved by an interesting inductive argument. This contradiction proves the theorem.

We give another proof for the exponential gap between k- and (k-1)-round complexity.

Theorem 2: For all $k \geq 1$ and for infinitely many n with $k \leq n/(96 \log n)$ there exists $L_k^n \subseteq \{0,1\}^{2n}$ such that $L_k^n \in \mathrm{COMM}_k(k \log n) - \mathrm{COMM}_{k-1}(n/4k)$.

Theorem 1 and 2 are incomparable. On the one hand the gap of Theorem 2 is wider. As we remarked above, there is no such a large gap for L_k of Theorem 1. Also if we take in Theorem 1 k to be a function of n and consider $L_k \cap \{0,1\}^{2n}$, then Theorem 2 is meaningful for a wider range of k. On the other hand, while the languages of Theorem 1 are simple and constructive, those of Theorem 2 are nonconstructive. The proof of Theorem 2 is an existence proof. In addition, the two proofs are entirely different. We suggest as an open problem, to prove a wide gap (from logn to $c_k n$) for a constructive language.

The proof of Theorem 2 considers sets $L \subseteq \{0,1\}^{2n}$ described by $2^n \times 2^n$ matrices. Once a partition π of the input bits is given the set is fully described by a $0 - 1$ matrix $M(L,\pi)$ with 2^n rows and columns corresponding to the possible bit strings seen by I and II.

The proof of Theorem 2 considers a <u>fixed</u> partition of the input. This is justified by Theorem 3 stated below. Once a partition π is fixed we can assume without loss of generality that $\pi = \pi_0$ the natural partition that gives I the first half of the input. We call the matrix $M \equiv M(L,\pi_0)$ <u>the matrix that corresponds to</u> L and refer to L as <u>the language that corresponds to</u> M. The matrix representation is due to Yao [6].

The computation can be viewed as follows: two computers called ROW and COLUMN have to recognize L. Each computer has one half of the input. (ROW knows the row in the matrix and COLUMN knows the column.) They alternate sending messages (each one of them can start). Both computers know the matrix of L. At any stage, each $i \in \{ROW, COLUMN\}$ knows the subset S_i of inputs the other may still have. When one of them sends a message the other one, j, obtains information that enables him to make S_j smaller. In fact the possible messages j receives imply a partition of S_j. The computation terminates when one of them, say ROW, has S_{ROW} such that all the entries in the row ROW has and the columns of S_{ROW} are the same (0 or 1). Note, that the submatrix corresponding to the final S_{ROW} and S_{COLUMN} should have the same entries (zeros or ones), because all corresponding input pairs have the same communication.

We construct the languages L_k^n inductively by constructing the corresponding matrices M_k^n. The matrices M_k^n are derived from simple matrices. The latter are obtained by repeating b-ary representation of the numbers 1,2,..., ℓ times. (b and ℓ are carefully chosen parameters.) Then, all i's in these matrices are replaced by $\pi_i(M_{k-2}^n)$ where π_i is a "random" permutation. The resulting matrix is M_k^n.

In the proof we define a meaningful portion of M_j^n. Let P_j be the claim that the submatrix corresponding to S_{ROW} and

S_{COLUMN} contains a meaningful portion of M_j^n. Then we show inductively that if P_j holds, then after exchanging two messages P_{j-2} holds. The proof makes use of the randomness of π_i. Another way to look at it is that by some interesting counting arguments we show that there exist permutations $\pi_0,\dots,\pi_{b-1}$ such that the above holds. The proof terminates when we observe that a meaningful portion of M_1^n must contain zeros and ones.

Recently [9] Yao considered probabilistic protocols and proved an exponential gap between one- and two-round probabilistic protocols. It is an interesting open problem to prove a result similar to Theorem 1 or 2 for such protocols.

When we fix a permutation π, we speak of a protocol $\varphi = (\varphi_c,\varphi_r)$ (where c(r) stands for column (row)) for (L,π). Let $L_1 \subseteq \{0,1\}^{2n}$, $L_2 \subseteq \{0,1\}^{2m}$ with partitions π_1,π_2.

(a) $(L_1,\pi_1) \leq (L_2,\pi_2)$, if for each protocol (φ_c,φ_r) for (L_1,π_1) there exist functions $f_1,f_2\colon \{0,1\}^n \to \{0,1\}^m$ such that $(\varphi_c(f_1,\ ),\varphi_r(f_2,\ ))$ or $(\varphi_r(f_1,\ ),\varphi_c(f_2,\ ))$ is a protocol for (L_2,π_2).

(b) $(L_1,\pi_1) \cong (L_2,\pi_2)$ if $(L_1,\pi_1) \leq (L_2,\pi_2)$ and $(L_2,\pi_2) \leq (L_1,\pi_1)$.

Remarks: The two cases in (a) mean that we treat symmetrically ROW and COLUMN. If $(L_1,\pi_1) \cong (L_2,\pi_2)$, then any (L_1,π_1)

computation immediately translates into (L_2, π_2) computation and vice versa.

Theorem 3: For each language $L \subseteq \{0,1\}^{2n}$ and partition π there exists $L^* \subseteq \{0,1\}^{8n}$ such that

(a) there exists τ with $(L,\pi) \cong (L^*,\tau)$

(b) for all σ $(L,\pi) \leq (L^*,\sigma)$.

In other words, whenever lower and upper bounds are proved for a language L and a fixed partition π, then there exists another language L^* such that these bounds hold independently of the partition.

It is interesting to note that Theorem 3 does not hold for nondeterministic protocols, because Aho et al showed [8] that for fixed partition there is only a polynomial (square) difference between deterministic and nondeterministic protocols.

The proof of Theorem 3 uses again probabilistic arguments. The matrix $M^* = M(L^*,\tau)$ is obtained from $M \equiv M(L,\pi)$ by first duplicating a large number of times the rows and columns of M and then by choosing two random permutations and permuting the rows and the columns of the resulting matrix. To establish (b) one observes that there are two such permutations such that for any partition of the input bits, the corresponding matrix contains a full copy of M or of M^T. Note that the proof of Theorem 3 introduces additional nonconstructiveness to the languages of Theorem 2.

The second main result in [1] was showing that for any integer function f, $1 \leq f(n) \leq n$, $COMM(f(n)) - COMM(f(n)-1) \neq \emptyset$. Our last result is:

Theorem 4: For any integer function f, $1 \leq f(n) \leq n$, $COMM_1(f(n)) - NCOMM(f(n) - 1) \neq \emptyset$.

Corollary 1: $COMM(f(n)) - COMM(f(n)-1) \neq \emptyset$.

Corollary 2: $NCOMM(f(n)) - NCOMM(f(n) - 1) \neq \emptyset$.

Theorem 4 extends the result in [1] (Corollary 1) to nondeterministic protocols in the strongest way. There seems to be no way to change the direct proof of Corollary 1 in order to prove theorem 4. The proof of Theorem 4 is rather simple.

The structure of the paper is as follows: the proof of Theorem i, i = 1,2,3,4, appears in Section i.

1. The Proof of Theorem 1.

We assume $L_{k+1} \in COMM_k(n^{1/2}/(36k^4\log^3 n))$ and derive a contradiction. Let $\Delta = \{D_n\}$ be the corresponding k-round protocols that recognize L_{k+1}. Without loss of generality each computation contains exactly k exchanged messages: by adding two bits we can record the fact whether the input has been accepted, rejected or neither. This increases the communication complexity by a constant (2k).

The proof consists of three parts. We first define several constants and prove a relationship among them (Claim 1). Next we define a subset of the inputs, S, corresponding to certain graphs. Then we prove Lemma 1 from which the theorem follows.

We consider inputs of length $2n = m2^m$, n large enough, as will be explained below. We choose the constants a, r, p (an integer), α and β, t and s (an integer) in this order to satisfy

(1) $n^a = n^{1/2}/(36k^4\log^3 n)$,

(2) $a = \frac{1}{2} - \frac{1}{r} \quad (r = \frac{\log n}{3\log\log n + 2\log 6k^2})$,

(3) $p = \lceil r \rceil$,

(4) $\alpha = \frac{1}{2} - \frac{1}{2p}$

(5) $\beta = \log(3kp)$,

(6) $t = \lceil 2^{m\alpha-\beta} \rceil /2$, and

(7) $s = \lfloor t \rfloor$.

These constants have been chosen so that

Claim 1: If n is large enough, then $s > kn^a$.

Proof: By (2) and (3), $\frac{1}{2p} - \frac{\log p}{\log n} \geq \frac{1}{2r+2} - \frac{\log(r+1)}{\log n} >$ $\frac{\frac{1}{2}\log\log n + \log 6k^2}{\log n}$ if n is large enough. So

$$\alpha - a \geq \frac{1}{2p} > \frac{\log p}{\log n} + \frac{\frac{1}{2}\log\log n + \log 6k^2}{\log n} =$$

$\frac{\frac{1}{2}\log\log n + \log k + \beta + 1}{\log n}$ (by (2)-(5)) and $n^{\alpha-a} > (\log n)^{\frac{1}{2}} k2^{\beta+1}$.

Hence, if n is large enough, since $2n = m2^m$,

$2^{m\alpha} - 2^{\beta} > \frac{n^{\alpha}}{(\log n)^{\alpha}} > \frac{n^{\alpha}}{(\log n)^{1/2}} > 2^{\beta+1}kn^a$. Thus

$s \geq t - 1/2 = \lceil 2^{m\alpha-\beta}\rceil/2 - 1/2 \geq 2^{m\alpha-\beta-1} - 1/2 > kn^a$. □

Each input consists of 2^m blocks of length m which will be identified with the numbers $0,1,\ldots,2^m - 1$. Considering the protocol D_n: (π,ω), each computer I, II sees a part (possibly empty) of each block (according to π). We say that block i is <u>free</u> for one of the computers if it sees at least αm bits in it. (Without loss of generality αm is an integer.) Note that since $\alpha < \frac{1}{2}$, a block may be free for the two computers. Note also that there are at least $2^m/(p+1)$ blocks free for each computer (because otherwise the other computer would see more than $(2^m-2^m/(p+1))(m-\alpha m) = m2^{m-1} = n$ bits).

We now identify $k + 2$ <u>disjoint</u> sets of blocks B_i, $i = 0,1,\ldots,k+1$, $B_i \subseteq \{0,1,\ldots,2^m-2\}$ that satisfy

(i) $B_0 = \{0\}$.

(ii) $|B_1| = 1$.

(iii) $|B_i| = \lceil 2^{m-\beta} \rceil$ for $i = 2,\ldots,k+1$.

(iv) For $i = 1,\ldots,k+1$, i odd (even), the blocks in B_i are free for II (I).

A simple counting argument shows that this is indeed possible. We say that the blocks in B_j with odd (even) j belong to II(I). Clearly if a block belongs to I(II) then it is free for I(II). For each block that belongs to I (II) we choose αm bits that I (II) sees, call them <u>free bits</u>, and call the other <u>fixed bits</u>.

We now describe a subset of the possible inputs $S \equiv X_0 \cdot X_1 \ldots X_{2^m-1}$, specifying for each block b a set $X_b \subseteq \{0,1\}^m$ of possible inputs.

(a) X_0 contains the unique number in B_1.

(b) For $b \in B_i$ and $1 \leq i \leq k$ we define X_b as follows. There are $\lceil 2^{m-\beta} \rceil$ strings representing numbers in B_{i+1}. These are partitioned according to the fixed bits of block b into $2^{m-\alpha m}$ subsets. One of these subsets has at least $\lceil 2^{m\alpha-\beta} \rceil$ strings. We choose from one such subset $\lceil 2^{m\alpha-\beta} \rceil$ elements to form X_b. Note that the so called fixed bits have fixed values in X_b.

(c) If $b \in B_{k+1}, X_b = \{1^m, y_b\}$, y_b contains 1's (0's) in the fixed (free) bits of the block b.

(d) If $b \notin \cup_i B_i$, $X_b = \{0^m\}$ (any fixed value will do).

With the graph interpretation in mind, ignoring the blocks not in $\cup_i B_i$ we have restricted attention to the following inputs. The possible directed graphs have k + 3 layers (the first k + 2 correspond to $B_0, \ldots, B_{k+1}$). Layers 0,1 and k + 2 contain one vertex (= block). Layer 0 contains block 0, and is connected (by an outoing edge) to layer 1. Layer k + 2 contains block $2^m - 1$. Layer i, $2 \leq i \leq k + 1$, contains $\lceil 2^{m-\beta} \rceil$ vertices. Each vertex in layers $1, \ldots, k$ is connected to one of $\lceil 2^{m\alpha-\beta} \rceil$ specific vertices in layers i + 1. Vertices in layer k + 1 are connected to one of two vertices, exactly one of which is the one in layer k + 2.

Moreover, for $1 \leq i \leq k$ i odd (even) II (I) has the entire information on layer i, because for each block in B_i he has the free bits. I (II) has no information at all on layer i because all fixed bits in B_i have the same value in X_b. To each input x in S corresponds a directed path that starts at vertex 0, goes through layers $1, 2, \ldots, k+1$ and either terminates in layer k + 2, in which case $x \in L$ or not, in which case $x \notin L$.

From now on we consider only inputs in S. For

$i = 1,2,\ldots,k+1$, let P_i be the possible input segments in the blocks of B_i (the marked concatenation of X_b for b in B_i). An element of S is represented by an element of $P_1 \times P_2 \times\ldots\times P_{k+1}$. For convenience we also include P_{k+2} which is the set containing the empty string.

We describe below a process that chooses in turn values from $P_1, P_2, \cdots$. After i stages, values from $P_1, \ldots, P_{i-1}$ have already been chosen and the value from P_i is restricted to one of s possible values $\{w_i^1, \ldots, w_i^s\}$. The input will be determined once an element of $P_i \times P_{i+1} \times\ldots\times P_{k+1}$ is chosen. If layer i belongs to computer I (say), then the input is determined once one of the s values of P_i as well as an element of $P_{i+2} \times P_{i+4}\ldots$ and an element of $P_{i+1} \times P_{i+3} \times\ldots$ are chosen. The first two values are known to I, while the third is known to II. While fixing values in P_j, $j = 1,2,\ldots$ we also restrict in a special way the possible continuations. After stage i only values from $V_i \subseteq P_{i+1} \times P_{i+3} \ldots$ are allowed for II and only values from $\cup_{j=1}^{s}\{w_i^j\} \times W_i^j \subseteq P_i \times P_{i+2} \times\ldots$ are allowed for I. Note that after stage i, all inputs that are still considered have the same corresponding initial path $g_0 = 0, g_1, \ldots, g_i$. The choice of w_i^j, $j = 1,\ldots,s$ will guarantee that g_i is connected to s possible vertices in layer $i + 1$ $\{g_{i+1}^j \mid j = 1,\ldots,s\}$. Lemma 1 describes this process more precisely. (Recall s and t of (6) and (7).)

Lemma 1: For each $i = 1,\ldots,k$ we can choose one value y_i from $P_1 \times \ldots \times P_{i-1}$, one vertex g_i in layer i, s possible values $w_i^1,\ldots,w_i^s$ from P_i, s different vertices $g_{i+1}^1,\ldots,g_{i+1}^s$ in layer $i+1$, subsets of values $V_i \subseteq P_{i+1} \times P_{i+3} \times \ldots$ and $W_i^j \subseteq P_{i+2} \times P_{i+4} \times \ldots$ for $j = 1,\ldots,s$, and a messages $c_i \in \{0,1\}^*$ such that:

(a) for $j = 1,\ldots,s$ all inputs in S represented by $(y_i, w_i^j \times W_i^j, V_i)$ contain the path $0, g_1,\ldots,g_i,g_{i+1}^j$, and correspond to the same (initial) computation $c_1 \$ c_2 \$ \ldots \$ c_i$ (independently of j).

(b) $|V_i| \geq (|P_{i+1}||P_{i+3}|\ldots)/(2^{n^a})^i$ for $i = 1,\ldots,k$; and $|W_i^j| \geq (|P_{i+2}||P_{i+4}|\ldots)/(2^{n^a})^i$ for $i = 1,\ldots,k-1$ and $j = 1,\ldots,s$.

Note that (b) means that the set of inputs still considered contains a large enough portion of all possible continuations for I and for II and for each choice of vertex in the next layer (the choice of g_{i+1} determined by that of w_i^j).

Proof: Induction on i.

Base: For $i = 1$, g_1 is the block in B_1, $w_i^1,\ldots,w_i^s$ are any s elements of P_1. The messages sent from I to II in the first round imply a partition of the possible inputs for I.

We choose a message c_1 with the largest corresponding part V_1. So, by (1), $|V_1| \geq (|P_2||P_4|\ldots)/2^{n^a}$. II can still have all inputs represented by $P_3 \times P_5 \times \ldots$, so the second half of (b) is immediate.

Induction step: Assume the lemma holds for $i \leq k - 1$. Let $q = \lceil |P_{i+1}|/(2(2^{n^a})^i) \rceil$. Consider $V_i = \cup_j \{u_j\} \times U_j$, $u_j \in P_{i+1}, U_j \subseteq P_{i+3} \times P_{i+5} \ldots.$ U_j is said to be large if $|U_j| \geq (|P_{i+3}| \cdot |P_{i+5}|\ldots)/(2(2^{n^a})^i)$.

Claim 2: For $i < k - 1$, there are at least q large U_j's.

Proof: Otherwise, if there were only $q' < q$ large U_j's, then

$$|V_i| \leq (q'|P_{i+3}| \cdot |P_{i+5}|\ldots) +$$
$$+ (|P_{i+1}| - q')(|P_{i+3}| \cdot |P_{i+5}|\ldots)/2(2^{n^a})^i <$$
$$(|P_{i+1}| \cdot |P_{i+3}|\ldots)/(2^{n^a})^i \leq |V_i|, \text{ contradiction.} \quad \Box$$

So, for $i < k - 1$ we can assume that $U_1, \ldots, U_q$ are large. If $i = k - 1$ we arbitrarily choose $u_1, \ldots, u_q$ from P_{i+1} and set $U_1 = \ldots U_q = \{\text{empty string}\}$.

Claim 3: There is an ℓ, $1 \leq \ell \leq s$, such that there are at least s different edges from g^{ℓ}_{i+1} to vertices in layer $i + 2$, when inputs from $y_i \times (\{w^{\ell}_i\} \times W^{\ell}_i) \times \cup^q_{j=1}\{u_j\} \times U_j$ are considered.

Proof: Assume to the contrary that for each ℓ, $1 \leq \ell \leq s$

the number of such edges is smaller than s. Hence the number of possible u_j's $q < s^s(\lceil 2^{m\alpha-\beta}\rceil)^{\lceil 2^{m-\beta}\rceil - s}$ $= s^s(2t)^{\lceil 2^{m-\beta}\rceil - s} \leq (2t)^{\lceil 2^{m-\beta}\rceil / 2^s} < (2t)^{\lceil 2^{m-\beta}\rceil / (2(2^{n^a})^i)}$ $= |P_{i+1}|/(2(2^{n^a})^i) \lesssim q$, a contradiction. (The last inequality is by Claim 1.) $\square$

To complete the proof of the lemma we use Claim 3: we choose $y_{i+1} = y_i \times w_i^\ell$, $g_{i+1} = g_{i+1}^\ell$, and for $j = 1,\ldots,s$, $w_{i+1}^j = u_j$ and g_{i+2}^j is determined by u_j. The s edges correspond to s elements of $\{u_1,\ldots,u_q\}$. Without loss of generality let them be $\{u_1,\ldots,u_s\}$. If $i < k - 1$ we choose $W_{i+1}^j = U_j$ for $j = 1,\ldots,s$. Since U_j is large, by Claim 1 the second part of (b) holds. The (i+1)-st message partitions W_i^ℓ into at most 2^{n^a} parts (by (1)). (Once $j = \ell$ is chosen W_i^ℓ represents the set of inputs for the computer which "owns" layers $i + 1$, $i + 3,\ldots$). Let V_{i+1} be the largest part. Hence $|V_{i+1}| \geq |W_i^\ell|/2^{n^a}$ and consequently (a) and the first part of (b) hold for $i + 1$. $\square$

It follows from Lemma 1, that all inputs represented by (y_k, w_k^j, V_k) for any j correspond to the same (complete) computation $c_1 \$ \ldots \$ c_k$. Hence the computer that receives the last message either accepts all of them, or rejects all of them. Hence, all these inputs must agree in all the blocks be $\{g_{k+1}^j,\ j = 1,\ldots,s\}$ (either all 0^m or all 1^m). But this would imply that $|V_k| \leq 2^{\lceil 2^{m-\beta} \rceil - s} < |P_{k+1}|/(2^{n^a})^k \leq |V_k|$. (The first inequality by Claim 1 and the second by Lemma 1.) The contradiction completes the proof of Theorem 1.

2. Proof of Theorem 2.

For $k = 1,3,5,\ldots$ we will define L_k^n for infinitely many n with $k \leq n/(36 \log n)$. We do it by defining the corresponding $m \times m$ 0-1 matrix, $m = 2^n$, $\hat{M}_k^m$. We will prove:

Lemma 2: (a) If COLUMN starts a k-round communication, then at most $k \log n$ bits need to be exchanged for recognizing L_k^n.

(b) If ROW starts a k-round communication, then more than $n/4k$ bits need to be exchanged for recognizing L_k^n.

Theorem 2 follows from Lemma 2:

For odd k: Obviously by (a) $L_k^n \in \mathrm{COMM}(k \log n)$. On the other hand, if $L_k^n \in \mathrm{COMM}_{k-1}(n/4k)$, then considering the corresponding (k-1)-round protocol and whenever COLUMN starts, changing it so ROW sends first the empty message, we obtain a k-round protocol that violates (b).

For even k: Define L_k^n by the $2m \times 2m$ matrix $\begin{pmatrix} \hat{M}_{k-1}^m & 0 \\ (\hat{M}_{k-1}^m)^T & 0 \end{pmatrix}$, $2m = 2^n$. Obviously, $L_k^n \in \mathrm{COMM}_k(k \log n)$: COLUMN starts. If the input for ROW is in the top half, then even k - 1 round suffice (by (a)) and if it is in the bottom half, COLUMN sends the empty message and then (again by (a)) after additional k-1 rounds L_k^n is recognized. On the other hand, if $L_k^n \in COMM_{k-1}(n/4k)$, if ROW (COLUMN) starts we restrict attention to the top (bottom) half of the matrix and derive a contradiction by (b).

Next, we define for k odd an $m \times C_k$ 0-1 matrix M_k^m with $C_k \leq m$. $\hat{M}_k$ above will be obtained from M_k^m by adding to it $m - C_k$ zero columns.

The Matrices M_k^m for k odd.

We now define M_k^m for $k = 2t + 1$ and infinitely many values of m. The values of m are chosen as follows: we choose an integer ℓ large enough, and a power of two b such that

(8) $\ell^{32} \leq b \leq (2\ell-k)^{32}$, and then choose

(9) $m = b^{\ell}$.

M_k^m is an $m \times C_k^{\ell}$ matrix, where C_k^{ℓ} is defined below. It is constructed from copies of $M_{k-2}^{m/b}$ which in turn is constructed from copies of M_{k-4}^{m/b^2} which eventually is constructed from copies of M_1^{m/b^t}. The last one is constructed directly.

The j-th matrix $j = 1,\ldots,t$, $M_{k-2j}^{b^{\ell-j}}$, is defined by induction because (8) holds for ℓ and k replaced by $\ell - j$ and $k - 2j$ and the same b.

The numbers of columns of these matrices $C_1^{\ell-t},\dots,C_k^{\ell}$ are defined by

$$(10)\quad C_{-1}^{s-1} = \lfloor b^s/s \rfloor$$
$$C_j^s = sC_{j-2}^{s-1},\ s \text{ integer } j = 1,2,\dots\ .$$

M_k^m is constructed with the help of a simple matrix $M_b(m)$ of the same dimensions: Let $0 \leq i < m$, and let the b-ary representation of i be $c_1c_2\dots c_\ell$. The i-th row of $M_b(m)$ is $(c_1,\dots,c_1,c_2,\dots,c_2,\dots,c_\ell,\dots,c_\ell)$ where each c_j repeats $C_{k-2}^{\ell-1}$ times. By (10), $M_b(m)$ is indeed an $m \times C_k^{\ell}$ matrix. Columns $jC_{k-2}^{\ell-1},\dots,(j+1)C_{k-2}^{\ell-1} - 1$ are called the j-th <u>column block</u> of $M_b(m)$.

M_1^m is $M_2(m)$. So the rows of M_1^m correspond to the binary representations of $0,\dots,m-1$.

To define M_k^m we need b permutations $\pi_0,\dots,\pi_{b-1}$ of sets of size $C_{k-2}^{\ell-1}$ that have certain properties. We will show later that such permutations exist and for the time being we assume that they are given. Consider in a column block of $M_b(m)$ the set of d-entries, $0 \leq d < b$. These entries form an $(m/b) \times C_{k-2}^{\ell-1}$ submatrix. Replace it by the matrix $\pi_d(M_{k-2}^{m/b})$ of the same size (π_d permutes the columns of its argument). We do it for each column block and each d and obtain M_k^m. If C is a column block of $M_{k-2}^{m/b}$, then $\pi_d(C)$ is referred to as a column block of $\pi_d(M_{k-2}^{m/b})$.

Claim 4: $C_k^\ell < b^\ell$.

Proof: By (10) and by induction on j, $\ell-t \leq j \leq \ell$, $C^j_{k-2(\ell-j)} \leq b^j$. □

As a result M_k^m has no more columns than rows. Add to it enough zero columns to make it square, and let L_k^n be the language corresponding to this matrix. Part (a) of Lemma 2 is immediate: COLUMN sends in his turn a number of a column block (between 1 and ℓ) and ROW sends in his turn a digit (between 0 and b-1). We start with M^m_{2t+1} and after two rounds have essentially $M^{m/b}_{2t-1}$. The communication complexity is therefore bounded by $(t+1)\log \ell + t \log b < k \log n$. The rest of this section is devoted to proving part (b) of Lemma 2.

Claim 5: (a) $k \leq \ell/3$, (b) $(C_k^\ell)^{1/\ell} \geq \ell$.

Proof: (a) $k \leq n/(96 \log n) = \log m/(96 \log \log m) \leq$
$\ell \log b/96(\log \ell + \log \log b) \leq \ell/3$.

(b) By (10) $C^\ell_{2t+1} = \ell(\ell-1)\ldots(\ell-t+1)C_1^{\ell-t} \geq$
$\ell(\ell-1)\ldots(\ell-t+1)b^{\ell-t}/2 \geq \ell^\ell$. □

Two Technical Lemmas.

Assume that we consider a class of rows from M_k^m. How many of the inserted matrices of level $k - 2$ have relatively many rows in common with this class?

Lemma 3: Let $r = m^\epsilon$ numbers from $\{0,\ldots,m-1\}$ be given in b-ary representation. Then there exist one digit position having at least $b^{\varepsilon-1/\ell}$ digits occuring with frequency $\geq r/(2b^2)$ (i.e. each digit occurs in $\geq r/(2b^2)$ numbers).

Proof: Remove those numbers whose first digit has frequency $\leq r/(2b^2)$. At most r/2b numbers are removed. Repeating this process for each digit position, we therefore remove at most r/2 numbers. The remaining numbers have only digits with frequency $\geq r/2b^2$.

Now let c_i be the number of digits in digit position i occuring in one of the remaining numbers. Then we have $c_1 \cdot c_2 \ldots c_\ell \geq r/2 = b^{\ell\varepsilon}/2$. So, there is one c_i with $c_i \geq (r/2)^{1/\ell} = b^{\varepsilon}/2^{1/\ell} \geq b^{\varepsilon-1/\ell}$. □

Consider next a class C of columns from M_k^m. Is there one inserted matrix of level k - 2, such that each of its column blocks has relatively many columns in common with C? The anwer is yes, provided we select the appropriate permutations:

Lemma 4: Let c be an integer, $c \geq 4\ell^3$. Given $B = B_1 \cup \ldots \cup B_\ell$, where $|B_i| = x$. [Interpret B as the set of columns in a column block of level k, B_i as those in the i-th column block of level k - 2.] Let p(y) denote the probability, that y randomly chosen permutation $\pi_1, \ldots, \pi_y$ of B have the property

(11) For each subset C of B of size c, there exists a permutation π_j, such that $|(\pi_j(B_i) \cap C)| \geq c/2\ell^2$ for <u>all</u> $1 \leq i \leq \ell$.

Then $P(4\ell^2 \log \ell x + y_0) \geq 1 - 2^{-y_0/4\ell^2}$.

<u>Proof:</u> Given C and B_i we count the number $n_{C,B_i}(d)$ of permutations π with

$$(12) \quad |\pi(B_i) \cap C| = d.$$

We have $n_{C,B_i}(d) = \binom{c}{d}\binom{\ell x - c}{x - d}x!(\ell x - x)!$ So the probability for (12) is $P_{C,B_i}(d) = \binom{c}{d}\binom{\ell x - c}{x - d}/\binom{\ell x}{x}$, and we have the hypergeometrical distribution, and

$$(13) \quad \frac{P_{C,B_i}(d-1)}{P_{C,B_i}(d)} \leq \frac{P_{C,B_i}(d)}{P_{C,B_i}(d+1)}$$

is valid for all d, and

$$(14) \quad P_{C,B_i}(c/\ell^2 - 1) \leq P_{C,B_i}(c/\ell^2)/\ell.$$

By (13) and (14) we conclude that $P_{C,B_i}(c/2\ell^2) \leq P_{C,B_i}(c/\ell^2)/\ell^{c/2\ell^2} \leq \ell^{-c/2\ell^2}$. If (11) does not hold, then for some C with size c, and all the y permutations (12) holds for some $1 \leq i \leq \ell$ and $0 \leq d < c/2\ell^2$ and all subsets C of size c. Hence, $1 - p(y) \leq \binom{\ell x}{c}[(c/2\ell)\ell^{-c/2\ell^2}]^y$. But since $c > 4\ell^3$, $c/2\ell < \ell^{c/4\ell^2}$ and hence $1 - p(y) \leq 2^{c \log((\ell x/c)\exp(1))}\ell^{-cy/4\ell^2}$. Thus, for $y = 4\ell^2 \log \ell x + y_0$, we have

$1 - p(y) \leq \ell^{-cy_0/4\ell^2}$. □

Corollary: For m,b,ℓ satisfying (8)-(10), and integer $c \geq 4\ell^3$ there exist b column permutations $\pi_1,\ldots,\pi_{b-1}$, such that: for each class C of c columns of

$$A \equiv \begin{bmatrix} \pi_1(M_{k-2}^{m/b}) \\ \\ \pi_b(M_{k-2}^{m/b}) \end{bmatrix}$$ and for each subset U of $\{1,\ldots,b\}$ of size $b^{1/8}$, there exists $i \in U$ such that each column block of $\pi_i(M_{k-2}^{m/b})$ has at least $c/2\ell^2$ columns in common with C. (Considering only C, there will be one relatively undamaged level k-2 matrix.)

Proof: Let B be the set of columns of A,

and let B_i be the set of columns of the i-th column block.

Then C is a subset of B of size $\geq 4\ell^3$. Let $\pi_1,\ldots,\pi_b$ be randomly chosen permutations of B. Given $U \subset \{1,\ldots,b\}$, $|U| = b^{1/8}$, the probability P_U, that $(\pi_u | u \in U)$ does not have the property (11) stated in Lemma 4 is bounded by $2^{-c(b^{1/8}-4\ell^2 \log \ell x)/2\ell^2}$. Since $b^{1/8} \geq \ell^4 \geq 4\ell^2 \log \ell x + \ell^4/2$, we have $P_U \leq 2^{-c\ell^4/4\ell^2} = 2^{-c\ell^2/4}$. So the probability P, that $\pi_1,\ldots,\pi_b$ don't have the property claimed in the corollary is bounded by $\binom{b}{b^{1/8}} 2^{-c\ell^2/4} \leq 2^{b^{1/8}\log(\exp(1)b^{7/8})-c\ell^2/4}$. But $b^{1/8} \log(\exp(1)b^{7/8}) - c\ell^2/4 \leq (2\ell)^4 \log(\exp(1)\cdot b) - \ell^5 < 0$

for ℓ large enough. Hence $P < 1$. □

The corollary defines the b permutations used in constructing M_k^m. It also specifies how large ℓ has to be.

Two Additional Lemmas.

A submatrix A of M_k^m is said to be undistinguishable after the i-th round, if for all rows and columns of A as possible inputs for ROW and COLUMN the same first i messages are exchanged.

A submatrix B of M_k^m is called a ε-fragment of M_k^m if B has a least m^ε rows and there are at least $(C_{k-2}^{\ell-1})^\varepsilon$ columns in B from every column block of M_k^m.

Lemma 5: Let B be an undistinguishable $(1-\varepsilon_0)$-fragment of M_k^m. Assume that ROW sends messages of length $< \delta \log m$ which are answered by messages from COLUMN of length $<$ $\delta \log C_{k-2}^{\ell-1}$. Then there exists one undistinguishable $(1 - \varepsilon_0 - \delta - 3/\ell)$-fragment of $M_{k-2}^{m/b}$ after these 2 rounds, provided $1 - \varepsilon_0 - \delta - 3/\ell > 1/8$.

Proof: First partition B according to the row messages. We concentrate on the largest class B_1 which must contain at least $m^{1-\varepsilon_0-\delta}$ rows. Applying Lemma 3 we find: there exists one column block (= digit position), B_1, where at least

$b^{\varepsilon-1/\ell}$ ($\varepsilon = 1 - \varepsilon_0 - \delta$) inserted matrices (= digits) have $m^{\varepsilon}/2b^2$ rows each (= frequence $\geq m^{\varepsilon}/2b^2$). But *all* column blocks of B_1 have at least $[C^{\ell-1}_{k-2}]^{1-\varepsilon_0}$ columns in each block. So B_1 has *one* column block with

1) at least $b^{\varepsilon-1/\ell}$ inserted matrices having $m^{\varepsilon}/2b^2 \geq m^{\varepsilon-3/\ell}$ rows each and
2) $[C^{\ell-1}_{k-2}]^{1-\varepsilon_0}$ columns.

Next, partition this column block according to the column messages. Again we take the largest class, which will have at least $(C^{\ell-1}_{k-2})^{1-\varepsilon_0-\delta}$ many columns. Call this class C. Now apply the corollary to Lemma 4. We get a permuted $M^{m/b}_{k-2}$ matrix with $m^{\varepsilon}/2b^2$ rows and $(C^{\ell-1}_{k-2})^{\varepsilon/(\ell-1)^2}$ $\geq (C^{\ell-1}_{k-2})^{\varepsilon-2/\ell}$ (by Claim 5 (b)) $\geq (C^{\ell-2}_{k-4})^{\varepsilon-3/\ell}$ columns per column block. Therefore we have an undistinguishable $(\varepsilon - \frac{3}{\ell})$-fragment of $M^{m/b}_{k-2}$. □

Lemma 6: Let $\varepsilon > 1/8$.Every ε-fragment of $M_1^{b^{\ell-t}}$ contains zeros and ones.

Proof: Interpret the $b^{(\ell-t)\varepsilon}$ rows as numbers in binary representation. Since $\varepsilon > 0$ at least two numbers occur. But then there must exist one digit position with both digits occuring in one of these numbers. So, we must have a column with both zeros and ones. □

The Proof of Lemma 2: Assume ROW starts a k-round protocol, $k = 2t + 1$, which exchanges $n/4k$ bits. Applying Lemma 5 t times with k and ℓ replaced by $k - 2j$, $\ell - j$, $j = 1,\ldots,t$, we find that M_1^{m/b^t} has an ϵ-fragment which is undistinguishable, $\epsilon \geq 1 - t/4k - 3t/(\ell-t)$. Since $k = 2t + 1$, and by Claim 5 (a), we have $\epsilon > 1/8$. But this contradicts Lemma 6. □

3. Proof of Theorem 3:

Let $M = M(L,\pi)$. Without loss of generality $\pi = \pi_0$, the natural partition. $M = (c_1,\ldots,c_{2^n})$, where $c_i \in \{0,1\}^{2^n}$ is the i-th column of M. Let $M_1 = (c_1,\ldots c_1,\ldots,c_{2^n},\ldots,c_{2^n})$, where each c_i is repeated 2^{3n} times. $M_1^T = (r_1,\ldots,r_{2^n})$, where $r_i \in \{0,1\}^{2^{4n}}$ is the i-th row of M_1. Let $M_2^T = (r_1,\ldots,r_1,\ldots,r_{2^n},\ldots,r_{2^n})$, where each r_i is repeated 2^{3n} times.

We say that two matrices A,B are equivalent if there are permutation matrices P,Q, such that $B = PAQ$. We prove below:

Lemma 7: There is a matrix M^* equivalent to M_2, such that the language L^* corresponding to M^* satisfies the following property: For each partition δ, one of the matrices M, M^T is equivalent to a submatrix of $M(L^*,\delta)$.

We consider M and M^T in the lemma because we will allow either ROW or COLUMN to start the computation. Lemma 7 establishes part (b) of Theorem 3. Part (a) is immediate with $\tau = \tau_0$ because of the way M^* was obtained from M. Lemma 7 makes use of Claim 6 below.

<u>Claim 6:</u> Let $\mathcal{G} = \{1,\ldots,2^{4n}\}$ and let $\mathcal{X}_1,\ldots,\mathcal{X}_p$, $p = \binom{4n}{2n}$ be p different subsets of $\mathcal{G}$ of size 2^{2n}, and let $\mathcal{G} = \mathcal{G}_1 \cup \mathcal{G}_2 \cdots \mathcal{G}_{2^n}$ be a partition of $\mathcal{G}$ into disjoint sets of size $_{2}3n$. Then, there is a permutation σ of $\mathcal{G}$ such that for all i,j, $1 \leq i \leq p$, $1 \leq j \leq 2^n$

$$(15) \quad \sigma(\mathcal{X}_i) \cap \mathcal{G}_j \neq \emptyset.$$

<u>Proof</u>: Let $P_{i,j}$ be the number of permutations of $\mathcal{G}$ that violate (15).

$$\frac{P_{i,j}}{(2^{4n})!} = \binom{2^{4n} - 2^{2n}}{2^{3n}} 2^{3n}!\,(2^{4n} - 2^{3n})!/(2^{4n})!$$

$$= \frac{(2^{4n}-2^{3n})\ldots(2^{4n}-2^{3n}-2^{2n}+1)}{2^{4n}\ \ldots\ (2^{4n}-2^{2n}+1)} \leq (1-2^{-n})^{2^{2n}}$$

$$\leq \exp(-2^n).$$

Hence $\Sigma_{i,j} P_{i,j} < (2^{4n})!$ $\square$

Let A be a $2^{4n} \times 2^{4n}$ matrix. Let $\underline{i} = \{i_1,\ldots,i_{2n}\}$ and $\underline{j} = \{j_1,\ldots,j_{2n}\}$ two sets of distinct integers between

1 and 4n. $A[\underline{i},\underline{j}]$ is the $2^{2n} \times 2^{2n}$ submatrix of A that consists of all rows (columns) of A with numbers that are obtained by taking 4n bit strings,assigning bits $i_1,\ldots,i_{2n}$ $(j_1,\ldots,j_{2n})$ in all possible ways and all other bits setting to 0. The rows (columns) of A which appear in $A[\underline{i},\underline{j}]$ are called the rows (columns) corresponding to $\underline{i}$ $[\underline{j}]$.

Proof of Lemma 7: Let P and Q be permutation matrices that correspond to permutations σ and τ defined later. Let $M^* = PM_2Q$ and let L^* be the corresponding language. Finally, let δ be any partition of $\{1,\ldots,8n\}$. Since we are looking for either M or M^T in $M(L^*,\delta)$, we can assume that COLUMN and ROW retain at least half of their input bits (according to π_0). So, bits $i_r (j_r)$ $r = 1,\ldots,2n$ of ROW (COLUMN) under π_0 are bits i'_r (j'_r) of COLUMN (ROW) under δ. Let $\underline{i} = (i_1,\ldots,i_r)$; $\underline{j}$, $\underline{i}'$, $\underline{j}'$ are defined similarly. Now

$$M(L^*,\delta)[\underline{i}',\underline{j}'] = M^*[\underline{i},\underline{j}] = M_2[\sigma(\underline{i}),\tau(\underline{j})].$$

The first equality holds because all the bits that belong to COLUMN (ROW) under π_0 and belong to ROW (COLUMN) under δ were set to 0. So, to prove the lemma it suffices to show that there exist σ and τ such that for all possible choices of $\underline{i}$ and $\underline{j}$ M is equivalent to a submatrix of

$M_2[\sigma(\underline{i}),\bar{\tau}(\underline{j})]$. This follows from Claim 6 as we now show.

Let $\mathcal{G}$ be the set of rows (columns) of M_2. Let $\mathcal{X}_1,\ldots,\mathcal{X}_p$ be the $p \equiv \binom{4n}{2n}$ possible row sets (column sets) of $M_2[\underline{i},\underline{j}]$ which correspond to the choices of $\underline{i}(\underline{j})$. $\mathcal{G} = \mathcal{G}_1 \cup \ldots \mathcal{G}_{2^n}$ is the partition of $\mathcal{G}$ to classes of copies of rows (columns) of M. The conclusion of the claim implies that there is a permutation $\sigma(\tau)$ such that for every possible choice of $\underline{i}(\underline{j})$, if we permute the rows (columns) of M_2 by $\sigma(\tau)$, we still have at least one copy of every row (column) of M among the rows (columns) of M_2 that correspond to $\sigma(\underline{i})$ $(\tau(\underline{j}))$. □

4. Proof of Theorem 4:

Claim 7: The class $COMM_1(f(n))$ contains at least $2^{2^{n+f(n)}}$ different subsets of $\{0,1\}^{2n}$ for every n and every integer function f with $0 \leq f(n) \leq n$.

Proof: Fix a string $y \in \{0,1\}^{n-f(n)}$. There are $2^{2^{n+f(n)}}$ different subsets of $\{0,1\}^{2n}$ of the form $\{xy|x \in \{0,1\}^{n+f(n)}\}$. Each one of them is in $COMM_1(f(n))$. □

We consider the following two cases.

Case 1: $f(n) \geq \log n$.

Let $L \in NCOMM(f(n))$ and let $D_n = (\pi,\varphi)$ be a nondeterministic protocol recognizing $L^n \equiv L \cap \{0,1\}^{2n}$

with communication complexity $f(n) - 1$. Let $c_1, c_2, \ldots, c_p$ be all the computations of length at most $f(n) - 1$ corresponding to D_n which end with accept. By the prefix freeness property $p \leq 2^{f(n)-1}$.

For $i = 1, \ldots, p$, let X_i^I (X_i^{II}) be the set of inputs that computer I (II) sees and correspond to the computation c_i. There is a one to one correspondence between L^n and $\cup_{i=1}^p X_i^I \times X_i^{II}$. This correspondence is determined by the partition π. Therefore, the number of such L^n is at most $\binom{2n}{n}\Sigma_{p=1}^{2^{f(n)-1}} \binom{(2^{2^n})^2}{p} \leq 2^{2n} 2^{f(n)-1} \binom{2^{2^{n+1}}}{2^{f(n)-1}} \leq$ $\dfrac{2^{2n}(2^{2^{n+1}})^{2^{f(n)-1}}}{(2^{f(n)-2})!} \leq \dfrac{2^{2n}\, 2^{2^{n+f(n)}}}{(n/4)!} < 2^{2^{n+f(n)}}$. ($\binom{2n}{n}$ = the number of π's, $\Sigma_{p=1}^{2^{f(n)-1}} \binom{(2^{2^n})^2}{p}$ = the number of possible $\cup_{i=1}^p X_i^I \times X_i^{II}$.) By Claim 7, there is an L^n in $COMM_1(f(n)) - NCOMM(f(n)-1)$.

Case 2: $f(n) \leq \log n$.

For $x \in \{0,1\}^*$ let $h(x)$ be the number of zeros in $x \bmod 2^{f(n)}$, and let $L^n = \{x | x \in \{0,1\}^{2n}, h(x) = 0\}$. Obviously $L^n \in COMM_1(f(n))$.

Now assume $L^n \in NCOMM(f(n)-1)$ and let $D_n = (\pi, \phi)$ be a nondeterministic protocol recognizing it with communication complexity $f(n) - 1$. As in Case 1, define $c_1, \ldots, c_p$, $p \leq 2^{f(n)-1}$. Let $A_\ell = \{x | x \in \{0,1\}^n, h(x) = \ell \bmod 2^{f(n)}\}$. Obviously,

$x \in L^n$ iff for some ℓ, $x^I \in A_\ell$ and $x^{II} \in A_{2^{f(n)}-\ell}$. Since there are $2^{f(n)} > 2^{f(n)-1} \geq p$ A_ℓ's, there must be x and y in L^n with $x^I \in A_\ell$ and $y^I \in A_m$ with $\ell \neq m$ that correspond to the same computation c_i. But then also the z with $z^I = x^I$ and $z^{II} = y^{II}$ must be accepted by c_i while $z \notin L^n$. □

5. References.

[1] C.H. Papadimitriou and M. Sipser, Communication complexity, Proc. 14th Annual ACM Symp. on Theory of Computing, 330-337, 1982.

[2] C.D. Thompson, Area-time complexity for VLSI, Proc. 11th Annual ACM Symp. on Theory of Computing, 81-88, 1979.

[3] R.J. Lipton and R. Sedgewick, Lower bounds for VLSI, Proc. 13th Annual ACM Symp. on Theory of Computing, 300-307, 1981.

[4] A.C. Yao, The entropic liminations on VLSI computations, Proc. 13th Annual ACM Symp. on Theory of Computing, 308-311, 1981.

[5] H. Abelson, Lower bounds on information transfer in distributed computations, Proc. 19th Annual IEEE Symp. Foundations of Computer Science, 151-158, 1978.

[6] A.C. Yao, Some complexity questions related to distributive computing, Proc. 11th Annual ACM Symp. on Theory of Computing, 209-213, 1979.

[7] K.Mehlhorn and E.M. Schmidt, Las Vegas is better than determinism in VLSI and distributed computing (extended abstract), Proc. 14th Annual ACM Symposium on Theory of Computing, 330-337, 1982.

[8] A.V. Aho, J.D. Ullman and M. Yanakakis, On notions of information transfer in VLSI circuits, Proc. 14th Annual Symp. on Theory of Computing, 133-139, 1983.

[9] A.C. Yao, Lower bounds by probabilistic arguments, Proc. 24th Annual IEEE Symp. on Foundations of Computer Science, 420-428, 1983.

[10] J. Ja'Ja', V.K. Prasanna Kumar and J. Simon, Information transfer under different sets of protocols, to appear in SIAM J. on Computing.

[11] J. Ja'Ja' and V.K. Prasanna Kumar, Information transfer in distributed computing with applications to VLSI, JACM 31 (1984), pp. 150-162.