BRICS

Basic Research in Computer Science

On Data Structures and Asymmetric Communication Complexity

Peter Bro Miltersen
Noam Nisan
Shmuel Safra
Avi Wigderson

BRICS Report Series RS-94-41

ISSN 0909-0878 December 1994

See back inner page for a list of recent publications in the BRICS Report Series. Copies may be obtained by contacting:

BRICS
Department of Computer Science
University of Aarhus
Ny Munkegade, building 540
DK - 8000 Aarhus C
Denmark

Telephone:	+45 8942 3360
Telefax:	+45 8942 3255
Internet:	BRICS@brics.dk

BRICS publications are in general accessible through WWW and anonymous FTP:

`http://www.brics.dk/`
`ftp ftp.brics.dk (cd pub/BRICS)`

On Data Structures and Asymmetric Communication Complexity

(extended abstract)

Peter Bro Miltersen [*] Noam Nisan [†] Shmuel Safra [‡]
Avi Wigderson [§]

Abstract

In this paper we consider two party communication complexity when the input sizes of the two players differ significantly, the "asymmetric" case. Most of previous work on communication complexity only considers the total number of bits sent, but we study tradeoffs between the number of bits the first player sends and the number of bits the second sends. These types of questions are closely related to the complexity of static data structure problems in the cell probe model.

We derive two generally applicable methods of proving lower bounds, and obtain several applications. These applications include new lower bounds for data structures in the cell probe model. Of particular interest is our "round elimination" lemma, which is interesting also for the usual symmetric communication case. This lemma generalizes and abstracts in a very clean form the "round reduction" techniques used in many previous lower bound proofs.

[*] BRICS (Basic Research in Computer Science, Centre of the Danish National Research Foundation) and Department of Computer Science, University of Toronto. Correspondence to: Dep. Comp. Sci, U. of Toronto, 10 King's College Road, Toronto ON M5S 1A4, Canada. Email: pbmilter@cs.toronto.edu. Supported by a grant from the Danish Natural Science Research Council and by the ESPRIT II Basic Research Actions Program of the European Community under contract No. 7141 (project ALCOM II).

[†] Dept. of CS, Hebrew University, Jerusalem. Supported by BSF 92-00043 and by a Wolfson research award. Part of this work was done while visiting BRICS at the University of Aarhus.

[‡] Hebrew U. and Weizmann Institute. Part of this work was done while visiting BRICS at the University of Aarhus.

[§] Dept. of CS, Hebrew University, Jerusalem. Part of this work was done while visiting BRICS at the University of Aarhus.

1 Introduction

In Yao's model of two-party communication [Yao79], the complexity of a protocol is the total number of bits communicated between the two players. An additional complexity measure sometimes considered is the number of rounds of messages. In most applications of communication complexity, it is sufficient to consider these two measures.

An exception is *asymmetric* communication problems where the input of one player (Alice) contains much fewer bits than the input of the other player (Bob). A simple example is the membership problem $MEM_{N,l}$, where Alice gets $x \in U = \{0, \ldots, N-1\}$, Bob gets $S \subseteq U$ of size at most l, and the two players must decide if $x \in S$. It is easy to verify that the communication complexity of the problem is $\lceil \log N \rceil$, and the trivial one round protocol, where Alice sends her entire input to Bob, is optimal.

However, this does not tell us all there is to know about the game. What if Alice does not send her entire input, but only, say, $\sqrt{\log N}$ bits? Will Bob have to send his entire input, or will fewer bits do? In general, what is the necessary tradeoff between the number of bits Alice sends Bob and the number of bits that Bob sends Alice? Standard lower bound techniques such as the rank technique [MS82] and the "large monochrome submatrix technique" [Yao83] fail to answer these questions. Some tradeoffs for specific functions have been obtained [Mil94, Mil95], but no generally applicable method for showing them has previously appeared.

1.1 Asymmetric Communication and Data Structures

One motivation for studying asymmetric communication complexity is its application to data structures in the *cell probe* model. The cell probe model, formulated by Yao [Yao81], is a model for the complexity of static data structure problems. In a static data structure problem, we are given a domain D of possible data, a domain Q of possible queries, and a map f on $Q \times D$, where $f(q, d)$ is the answer to query q about data d. A solution with parameters s, b and t, is a method of storing any $d \in D$ as a data structure $\phi(d)$ in the memory of a random access machine, using s memory cells, each containing b bits, so that any query in Q can be answered by accessing at most t memory cells. We are interested in tradeoffs between s, the size of the data structure, and t, the query time (the value of b being regarded as a parameter of the model, usually $O(\log |Q|)$ or $O(\text{polylog } |Q|)$).

A familiar example is the (existential) two dimensional orthogonal range query problem, where D is the set of subsets $S \subset \{1, \ldots, N\} \times \{1, \ldots, N\}$ of a certain size, Q is the set of rectangles $[x, y] \times [z, u]$, and $f([x, y] \times [z, u], S) = 1$ if and only if $[x, y] \times [z, u] \cap S \neq \emptyset$.

It was observed in [Mil94] that lower bounds for cell probe complexity can be derived using communication complexity: For a static data structure problem, we consider the communication problem, where Alice gets $q \in Q$, Bob gets $d \in D$, and they must determine $f(q, d)$. If there is solution to the data structure problem with parameters s, b and t, then there is a protocol for the communication problem, with $2t$ rounds of communication, where Alice sends $\log s$ bits in each of her messages and Bob sends b bits in each of his messages. For natural data structure problems $|q| = \log |Q|$ is much smaller than $|d| = \log |D|$, so the communication problem is asymmetric. Earlier lower bounds for static data structures in the cell probe model [Ajt88, Xia92] also fit into the communication complexity framework.

In section 2 we continue studying the relations between complexity in the cell probe model and asymmetric communication complexity. We show that:

- When the number of rounds of communication is constant, the communication complexity also provides upper bounds for cell probe complexity.

However, by a result in [Mil93], when the number of rounds of communication is not constant, for almost all data structure problems (with natural choices of parameters) the cell probe complexity is significantly (as much as exponentially) larger than the communication complexity. This may suggest that the asymmetric communication complexity approach is not the best one for proving lower bounds in the cell probe model. However, our next result shows that obtaining better lower bounds, using any method, may be very difficult. The best bounds that can be obtained (and we do obtain) using communication complexity are $t = \Omega(n/\log s)$, where $n = \log |Q|$, and we show that much better lower bounds imply time-space tradeoffs for branching programs, a long standing open problem (see e.g. [Weg87], pp. 423).

- If a function $f : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}$ can be computed by polynomial size, read $O(1)$ times branching programs, then there is a data structure storing $d \in \{0,1\}^m$ using $s = m^{O(1)}$ cells of size b each so that any query $q \in \{0,1\}^n$ can be answered in $t = O(n/\log b)$ queries.

We go on to provide two generally applicable techniques for showing necessary tradeoffs between the number of bits that Alice sends, the number of bits that Bob sends, and the number of rounds of communication. We apply them to a variety of problems, some of them motivated by cell probe complexity, others by their intrinsic interest.

Some notation: Let $f : A \times B \to \{0,1\}$ be a communication problem.

An $[a,b]$-protocol for f is a protocol where the *total* number of bits that Alice sends Bob is at most a and the *total* number of bits that Bob sends Alice is at most b.

A $[t,a,b]^A$-protocol for f is a protocol where *each* of Alice's messages contains at most a bits and *each* of Bob's messages contains at most b bits and at most t messages are sent, with Alice sending the first message. A $[t,a,b]^B$-protocol is defined similarly.

A randomized protocol for f is a public coin protocol P where for every x, y, $\Pr(P(x,y) = f(x,y)) \geq 2/3$. It has one-sided error if $f(x,y) = 0 \Rightarrow \Pr(P(x,y) = 0) = 1$.

1.2 The Richness Technique

Our first general technique, presented in section 3, is the use of the following *richness lemma*. Identify f with its communication matrix M with $M_{a,b} =$

$f(a, b)$, i.e. index the rows by Alice's possible inputs, and the columns by Bob's possible inputs. We say that a matrix (and a problem) is (u, v)-*rich* if at least v columns contains at least u 1-entries.

Richness Lemma: Let f be a (u, v)-rich problem. If f has a randomized one-sided error $[a, b]$-protocol, then f contains a submatrix of dimensions at least $u/2^{a+2} \times v/2^{a+b+2}$ containing only 1-entries.

The lemma is easy to prove and simple to use, and it enables us to give good lower bounds for several problems.

- In the disjointness problem, Alice gets $T \subseteq \{0, \ldots, n-1\}$ of size k, Bob gets $S \subseteq \{0, \ldots, n-1\}$ of size l, and they must decide if $T \cap S = \emptyset$. (The symmetric version of this problem is, of course, well studied.) We prove that in any randomized one-sided error $[a, b]$ protocol either $a = \Omega(k)$ or $b = \Omega(l)$. Furthermore, if $k < a < k \log l$, then $b \geq l/2^{O(a/k)} - a$. We also provide non-trivial upper bounds.

- The membership problem is the interesting special case where $k = 1$. In this case our tradeoffs are particularly tight.

- In the span problem, Alice gets an n-dimensional vector $x \in {Z_2}^n$, and Bob gets a subspace $Y \subseteq {Z_2}^n$ (represented, e.g., by a basis of $k \leq n$ vectors). They must decide whether $x \in Y$. We show that essentially no non-trivial protocol exists: in any randomized one-sided error $[a, b]$ protocol either $a = \Omega(n)$ or $b = \Omega(n^2)$.

These communication complexity lower bounds have as direct corollaries lower bounds in the cell probe model regarding data structures maintaining subsets of of $\{1...n\}$, or subspaces of ${Z_2}^n$, respectively.

1.3 The Round Elimination Lemma

Our second technique, presented in section 4, is a round-by-round "restriction" of the protocol. These types of techniques lie at the heart of all previously known lower bounds for static data structures [Ajt88, Xia92, Mil94, BF94], and several other lower bounds in communication complexity [KW90, DGS84, HR88, NW93]. In each case they have been used in an ad-hoc way. We obtain a very general lemma abstracting these types of techniques.

Given f, we define a new communication problem as follows: Alice gets m strings $x_1, ..., x_m$ and Bob gets a string y and an integer $1 \leq i \leq m$. Their aim is to compute $f(x_i, y)$. Suppose a protocol for this new problem is given, where Alice goes first, sending Bob a bits, where a is much smaller than m. Intuitively, it would seem that since Alice does not know i, the first round of communication can not be productive. We justify this intuition. Moreover, we show that this is true even if Bob also gets copies of $x_1, ..., x_{i-1}$, a case which is needed in some applications. Denote this problem by $P_m(f)$.

Round Elimination Lemma: Suppose there is a randomized $[t, a, b]^A$-protocol for solving $P_{100a}(f)$. Then there is a randomized $[t-1, 120a, 120b]^B$-protocol for solving f.

This lemma can be applied to a wide range of problems with the following kind of "self reducibility": $P_m(f)$ (with given parameters) can be reduced to a single problem f (naturally with larger parameters). In these cases we can use the lemma repeatedly, each time shaving off another round of communication.

- Our first application is to obtain the first lower bounds for data structures for the existential two-dimensional orthogonal range query problem described above[1]. The lower bound applies also to higher dimensions, but for the one-dimensional problem we prove a constant upper bound. This answers questions raised in [Mil94].

We then demonstrate the power of the lemma by easily deriving (sometimes with somewhat weaker bounds) several of the known lower bounds both for data structure problems and for other communication complexity problems. These include:

- Lower bounds for data structures for counting and modulo-counting versions of the 1 dimensional range query problem in the cell probe model. Such bounds were first proved in [Ajt88, Xia92, Mil94, BF94].
- The depth hierarchy for monotone constant depth circuits. This was first proved by [KPPY84] and, using Karchmer-Wigderson games [KW90], is equivalent to a rounds problem in communication complexity (see [NW93]), which we prove a lower bound for.
- A round-communication tradeoff for the randomized complexity of the "greater than" problem. (Alice and Bob each get an n-bit integer and they must decide which is greater.) This was first proved by Yao (unpublished).

2 Communication Complexity vs. Cell Probe Complexity

Communication complexity is the only known generally applicable method for showing lower bounds on the cell probe complexity of static data structure problems. In this section we discuss how powerful it is, and the likelihood of more powerful methods.

Let a data structure problem f on domains $Q = \{0,1\}^n$ and $D = \{0,1\}^m$ be given. How large tradeoffs between structure size s and query time t can be shown?

In [Mil94] it was shown that the following communication complexity problem provides lower bounds for the query time. Alice gets $q \in Q$, Bob gets $d \in D$, and they must determine $f(q,d)$.

Lemma 1 *[Mil94] If there is solution to the data structure problem with parameters s, b and t, then there is a $[2t, \lceil \log s \rceil, b]^A$-protocol for the communication problem.*

[1] After first obtaining a lower bound using the round elimination lemma, we have discovered an alternative proof involving a simple reduction to "parity range query problems". This second proof also yields better lower bounds relying on the ones of [BF94].

We can provide a converse in the restricted case where the communication complexity protocol has a constant number of rounds.

Lemma 2 *If there is a $[O(1), a, b]$ protocol for computing $f(q, d)$ then the data structure problem has a solution with parameters $s = 2^{O(a)}$, $t = O(1)$, and b.*

Proof: (sketch) There will be a cell for any possible transmission by Alice. That cell will hold Bob's answer.

□

A more general converse is, however, impossible. Using communication complexity, we can at most show an $n/\log m$ lower bound, since in this number of rounds, Alice can send her entire query to Bob. However, there are well known data structure problem, where the best known upper bound on the query time is much larger than $n = \log|Q|$. A notoriously difficult example is the *partial match query* problem where we must store a subset $S \subseteq \{0,1\}^n$, so that for any $q \in \{0,1\}^n$, the query "$\exists z \in S \forall i : q_i \leq z_i$?" can be answered. No solution is known with worst case query time even polynomial in n when the struture size is polynomial. Yet not only does communication complexity fail to provide bounds better than $n/\log m$, but for this problem, we only know how to show a $\sqrt{\log n}$ lower bound, using the techniques of section 4. Counting arguments show that for most data structure problems the solution which stores the non-redundant representation of the data and the query algorithm which reads all of it, is in fact optimal:

Theorem 3 *[Mil93] For a* random *data structure problem $f : Q \times D \to \{0,1\}$, if $s < \log|Q|/2b$ cells of size b are used then query time $\Omega(\log|D|/b)$ is necessary.*

Thus, for a random function there is a huge (as much as exponential) gap between cell probe complexity and communication complexity. We don't know any explicitly defined function with a provable gap. Finding one is an interesting open problem. The following theorem tells us that we are unlikely to get superlinear (in n) lower bounds for explicitly defined functions with the current state of the art of complexity theory. Recall that it is still an open problem (believed to be difficult) whether all of *NP* can be computed by polynomial size, read twice branching programs (see e.g. [Weg87], pp. 423).

Theorem 4 *If a function $f : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}$ can be computed by polynomial size, read $O(1)$ times branching programs, then there is a data structure storing $d \in \{0,1\}^m$ using $s = m^{O(1)}$ cells of size b so that any query can be answered in time $t = O(n/\log b)$.*

Proof: Let us first show a data structure with a $O(n)$ upper bound on the query time, and thereafter show how to improve it to $n/\log b$.

Given a branching program for f of size $(n+m)^{O(1)} = m^{O(1)}$, and a data structure instance $d \in \{0,1\}^m$, eliminate all d_i-variables in the branching program, leaving only q_i-variables. The size has not increased. We store a pointer structure representing this new branching program.

Given a query q, we simulate the stored branching program on q. Since the branching program reads each variable only a constant number of times, the query time is $O(n)$.

We now present the improved version. Note that since the branching program has size m^c we only need $c \log m$ bits to represent pointers in the program. Thus, we can in a single cell represent a binary tree of depth $r \geq \log b/2$ with pointers to branching program locations in the nodes and indices of q_i-variables on the edges. For each branching program location, we make such a cell, representing the program for the next r steps. This speeds up simulation of the program with a factor r.

□

3 The Richness Technique

3.1 The Richness Lemma

Given a communication problem $f : A \times B \to \{0, 1\}$, we identify f with its communication matrix M with $M_{a,b} = f(a, b)$, i.e. we index the rows by Alice's possible inputs, and the columns by Bob's possible inputs. We say that a matrix (and a problem) is (u, v)-*rich* if at least v columns contain at least u 1-entries.

Lemma 5 *Let f be a (u, v)-rich problem. If f has a randomized one-sided error $[a, b]$-protocol, then f contains a submatrix of dimensions at least $u/2^{a+2} \times v/2^{a+b+2}$ containing only* 1*-entries.*

The proof is postponed to the appendix.

3.2 The membership problem

In the membership problem $MEM_{n,l}$, Alice gets $x \in \{0, 1, \ldots, n-1\}$, Bob gets $S \subseteq \{0, 1, \ldots, n-1\}$ of size at most l, and they must decide if $x \in S$. Assume for convenience that n and l are powers of two, and that $l \leq n/2$. Being asymmetric, this problem has not been studied previously. Let us first look at some upper bounds. Between the extreme behaviors of the $[1, l \log n]$-protocol, where Bob sends his entire input to Alice, and the $[\log n, 1]$-protocol where Alice sends his entire input to Bob, we have the following protocols.

Theorem 6 *The non-membership problem has the following protocols:*

1. *For $a \leq \log l$, a $[2a, O(l \log n/2^a)]$-protocol, and for $a \geq \log l$, a $[2a, O(\log n + 2\log l - 2a)]$-protocol.*

2. *For all $a \leq \log l$, a randomized one sided error $[O(a), O(l/2^a)]$-protocol.*

Proof: Deterministic Protocol: First consider $a \leq \log l$. Before the protocol starts, the two players agree on a prime p between n and $2n-1$. Consider the family of hashfunctions

$$h_k(x) = (kx \bmod p) \bmod 2^{2a-1}.$$

Bob chooses k so that the number of collisions of h_k on S is minimized. As shown in [FKS84], he can choose one so that the total number of collisions is at most $O(l^2/2^{2a})$. He sends it to Alice, who hashes her input and sends the result to Bob, who sends Alice all those elements $\in S$ with the same hash value. Note that if r elements have the same hash value, then the number of collisions is greater than $\binom{r}{2}$, so he sends at most $O(l/2^a)$ elements. Finally, Alice tells Bob if her input is among them.

For $a \geq \log l$, Alice reduces the domain size by sending Bob the first $a-2\log l$ bits of her input, after which they simulate the first protocol.

Randomized Protocol: This is just a special case of the randomized protocol for disjointness (lemma 8).

□

Note that all of the above protocols are constant round. We now use the richness lemma to show lower bounds.

Theorem 7 *If $MEM_{n,l}$ has a one-sided error $[a,b]$-protocol, then $2^a(a+b) = \Omega(l(\log n - \log l))$. If its negation has a one-sided error $[a,b]$-protocol, then $2^a(a+b) = \Omega(l)$.*

Proof: The (n,l)-membership function is $(l, \binom{n}{l})$-rich, so by the richness lemma, we can find a 1-submatrix of dimensions at least $l/2^{a+2} \times \binom{n}{l}/2^{a+b+2}$. Note, however, that if the membership matrix contains a 1-rectangle of dimensions $r \times s$, then $\binom{n-r}{l-r} \geq s$ so

$$\binom{n-l/2^a+2}{l-l/2^{a+2}} \geq \binom{n}{l}/2^{a+b+2} \Rightarrow 2^{a+b+2} \geq (n/l)^{l/2^{a+2}} \Rightarrow 2^{a+2}(a+b+2) \geq l(\log n - \log l)$$

The negation of $MEM_{n,l}$ is $(n-l, \binom{n}{l})$ rich, so by the richness lemma, we can find a 1-submatrix of dimensions $(n-l)/2^{a+2} \times \binom{n}{l}/2^{a+b+2}$. Note, however, that if the non-membership matrix contains a 1-submatrix of dimensions $r \times s$, then $\binom{n-r}{l} \geq s$, so

$$\binom{n-\frac{n-l}{2^{a+2}}}{l} \geq \binom{n}{l}/2^{a+b+2} \Rightarrow a+b+2 \geq l\log(\frac{n}{n-\frac{n-l}{2^{a+2}}}) \Rightarrow 2^a(a+b) = \Omega(l)$$

□

If we are only interested in the value of a and b up to a constant, the deterministic upper bounds and the lower bounds for one-side error protocols are tight for $l \leq n^{1-\epsilon}$ and $a \leq \log l$: It is sufficient and necessary that $b = l\log n/2^{\Theta(a)}$. The bounds for randomized one-sided error protocols for non-membership tight for any $l \leq n/2$ and a: It is sufficient and necessary that $b = l/2^{\Theta(a)}$.

3.3 The Disjointness Problem

An obvious generalization of the membership problem is the disjointness problem $DISJ_{n,k,l}, k < l < n/2$, where Alice gets $T \subseteq \{0,\ldots,n-1\}$ of size k, Bob gets $S \subseteq \{0,\ldots,n-1\}$ of size l, and they decide if $T \cap S = \emptyset$. The symmetric version of this problem is, of course, well studied.

Several upper bounds can be derived for this problem using extensions of the protocols given for the membership problem. Perhaps the nicest is:

Lemma 8 *$DISJ_{n,k,l}$, for $k < l < n/2$ has a one-sided error randomized $[O(a), O(l/2^{a/k})]$-protocol for all values of $k \leq a \leq k \log l$, and a one-sided error randomized $[O(a), O(l \log(k/a))]$ protoocl for all values of $1 \leq a \leq k$.*

Proof: We use an adaptation of a protocol due to Hastad and Wigderson (unpublished). First let us consider the $a = \Theta(k)$ case. Here the public coin flips will denote a sequence of random subsets $R_1 ... R_i ...$ of $\{1...n\}$. Each round Alice will send to Bob the next i such that $S \subseteq R_i$, Bob will update his set $T \leftarrow T - R_i$, and will send to Alice $j - i$ for the next j such that $T \subseteq R_j$ (the new T), and then Alice will update $S \leftarrow S - R_j$. If at any point during the protocol S or T become empty then the original sets were disjoint. The expected number of bits sent by Alice (resp. Bob) in each round is the current size of S (resp. T). If S and T are disjoint then the expected size of both S and T decreases by a factor of exactly 2 each round. Thus the total expected number of bits sent by Alice (resp. Bob) is still $O(k)$ (resp. $O(l)$). If S and T do not become empty after so many bits have been sent then, w.h.p, S and T were not disjoint.

If $a \geq k$ then Alice starts by sending Bob $O(a/k)$ indices i as before. This allows Bob to reduce the size of T (assuming that it is disjoint from S) by an expected factor of exactly $2^{a/k}$. Then they continue with the previous protocol. If $a \leq k$ then Bob starts by sending Alice $\log(k/a)$ indices i as before, reducing the size of S to $O(a)$.

□

Theorem 9 *If the disjointness problem has a randomized one-sided error $[a, b]$-protocol, then either $a = \Omega(k)$ or $b = \Omega(l)$. Moreover, for $a > k$, $b = \Omega(l/2^{a/k} - a)$*

Proof: The (n, l)-disjointness function is $(\binom{n-l}{k}, \binom{n}{l})$-rich, so by Lemma 5, we can find a 1-rectangle of dimensions at least $\binom{n-l}{k}/2^a \times \binom{n}{l}/2^{a+b}$. Let the rows be indexed by the sets $T_1, T_2, \ldots, T_r$ and let the columns be indexed by the sets $S_1, S_2, \ldots, S_s$. We then have that $S_i \cap T_j = \emptyset$ for all i, j. Let $t = (n-l-k)/2^{a/k}$. Since $\binom{t}{k} < \binom{n-l}{k}/2^a$, we must have $\cup T_i > t$ and therefore $\cup S_i < n - t$, i.e. $2^{a+b} \geq (\frac{n}{n-t})^l$.

□

The disjointness problem is interesting from a cell probe perspective. Recall that by perfect hashing [FKS84], one can store a set $S \subseteq U$ using $O(|S|)$ cells, each containing an element of U, so that membership queries can be answered in constant time. Now suppose we have k elements, and we want to find out whether any of them are in S. Is there a data structure for S and a way of preprocessing the query so that after preprocessing, we can do this in $o(k)$ time? As a corollary to the above theorem, we can show that there is not.

3.4 The Span Problem

The membership and disjointness problems exhibits a smooth tradeoff between the number of bits that Alice sends Bob and the number of bits that Bob sends

Alice. Using the richness technique, we can show that this is not the case for the problem INSPAN_n, where Alice gets $x \in {Z_2}^n$, Bob gets a vector subspace $Y \subseteq {Z_2}^n$, (the subspace may be represented by a basis of $k \leq n$ vectors, thus requiring $O(n^2)$ bits) and they must decide whether $x \in Y$. We omit the proof of the following theorem.

Theorem 10 *In any $[a, b]$ one-sided error randomized protocol for INSPAN_n either $a = \Omega(n)$ or $b = \Omega(n^2)$.*

4 The Round Elimination Technique

4.1 Round Elimination Lemma

Let $f(x, y)$ be a communication problem on domain $X \times Y$. Let $P_m(f)$ be the following problem: Alice gets m strings $x_1, ..., x_m \in X$; Bob gets an integer $i \in \{1..m\}$, a string $y \in Y$ and a copy of the strings $x_1, ..., x_{i-1}$. Their aim is to compute $f(x_i, y)$.

Lemma 11 *Suppose there is a randomized $[t, a, b]^A$-protocol for solving $P_{100a}(f)$. Then there is a randomized $[t-1, 120a, 120b]^B$-protocol for solving f.*

The proof of this main lemma is quite involved and is postponed to the appendix.

4.2 Range query problems

We consider the cell probe complexity of existential, r-dimensional orthogonal range query problems on domain $U = \{1, \ldots, 2^n - 1\}$, for fixed $r \geq 1$.

The problem is as follows: Given a data set $S \subseteq U^r$, construct a static data structure using at most $s = |S|^{O(1)}$ memory cells, each containing $b = n^{O(1)}$ bits, so that for any box $q = [u_1, v_1] \times \cdots [u_r, v_r]$, we can answer the query "Is $q \cap S = \emptyset$?" efficiently.

Previously, only *counting* range queries (where the query is "What is $|q \cap S|$?"), and *modulo-counting* range queries (where the query is "Is $|q \cap S| \bmod r = 0$?") have been considered in the cell probe model. An upper bound on the query time in the one-dimensional problem, for all types of queries, is $O(\log n)$, with $s = O(|S|), b = O(n)$ [Wil83]. It is easy to generalize this to a solution for the r-dimensional problem with query time $O(\log n)$ and $s = O(|S|^r), b = O(n)$. The best known lower bound for counting [Xia92] and modulo-counting [Mil94, BF94] range queries is $\Omega(\log n / \log\log n)$ for any dimension $r \geq 1$. ([Xia92] and [Mil94, BF94] were done independently). The complexity of existential queries was left as an open problem in [Mil94].

Here, we show a $O(1)$ *upper* bound on the query time for existential range queries in the one-dimensional case, and an $\Omega(\sqrt{\log n})$ lower bound on d-dimensional queries for $r > 1$.

For the upper bound, by Lemma 2, we only need to find a protocol for the communication problem $\mathit{OERQ}_{n,l}$, where Alice gets an interval $[q_1, q_2]$, Bob gets a set $S \subseteq U$ of size at most l and they have to decide if $[q_1, q_2] \cap S = \emptyset$.

Theorem 12 *$OERQ_{n,l}$ has an $[O(1), O(\log l), n^2]$-protocol.*

Proof: If $|S| \leq n$, Bob can send his entire input to Alice in one round, so assume $S > n$. Identify q_1 and q_2 with their binary representation, and let $i \in \{1, \ldots, n\}$ be the first bit where q_1 and q_2 differ, and let w be their common prefix of length $i-1$. Since $q_1 < q_2$, we have $q_{1i} = 0$ and $q_{2i} = 1$. We can write $[q_1, q_2] = [q_1, z-1] \cup [z, q_2]$, where $z = w10^{n-i-1}$. Our protocol determine if $[q_1, q_2] \cap S = \emptyset$ by checking if $[q_1, z-1] \cap S = \emptyset$ and if $[z, q_2] \cap S = \emptyset$. We only describe the second part, the first is similar.

Alice sends i to Bob using $\lceil \log n \rceil = O(\log l)$ bits. They now determine if there an element in S starting with the prefix $w1$. This is done by the deterministic $[O(\log l), O(\log n)]$-membership protocol of Section 3. If there isn't such an element $[z, q_2] \cap S$ is empty. Otherwise, the membership protocol also tells Bob exactly what w is, and he can send Alice the smallest of his element y with prefix $w1$. Alice then checks if q_2 is smaller than y, in which case $[z, q_2] \cap S$ is empty, otherwise it isn't. This completes the protocol.

□

We now turn to show lower bounds on r-dimensional queries for $r > 1$. We assume without loss of generality that $r = 2$, and consider the communication problem $ERQ_{n,l}$ where Alice gets $(x, y) \in U^2$, Bob gets $S \subseteq U^2$ of size at most l and they must determine if $([1, x] \times [1, y]) \cap S = \emptyset$. Our lower bound is an immediate corollary of the following theorem:

Theorem 13 *Let any $c > 1$ be given. For a sufficiently large n, let $l = 2^{(\log n)^2}, a = (\log n)^3, b = n^c, t = \sqrt{\log n}/10$. Then $ERQ_{n,l}$ does not have an $[t, a, b]$-protocol.*

Proof: For a communication problem f, let $P^m(f)$ be defined as $P_m(f)$ but with the roles of Alice and Bob reversed. The round elimination lemma enables us to reduce instances of ERQ to $P_m(ERQ)$ or $P^m(ERQ)$, eliminating one round. We also need to reduce instances of $P_m(ERQ)$ or $P^m(ERQ)$ to ERQ. The following two reductions take care of that:

Suppose that m divides n. A communication protocol for $ERQ_{n,l}$ can be used as a protocol for $P_m(ERQ_{n/m,l})$ as follows: Alice, given $(x_1, y_1) \ldots, (x_m, y_m)$, computes the concatenations $x' = x_1 \cdot x_2 \cdots x_m$ and $y' = y_1 \cdot y_2 \cdots y_m$. Bob, given i, S, and $(x_1, y_1), \ldots, (x_{i-1}, y_{i-1})$ computes $S' = \{(x_1 \cdot x_2 \cdots x_{i-1} \cdot u,\ y_1 \cdot y_2 \cdots y_{i-1} \cdot v) | (u, v) \in S\}$. Since $[1, x'] \times [1, y'] \cap S' = \emptyset$ iff $[1, x_i] \times [1, y_i] \cap S = \emptyset$, they get the correct result by simulating the $ERQ_{n,l}$ protocol.

Suppose m is a power of two. A communication protocol for $ERQ_{n,l}$ can be used as a protocol for $P^m(ERQ_{n-\log m, l/m})$ as follows: Alice, given (x, y) and i, computes $x' = [i-1] \cdot x$ and $y' = [n-i] \cdot y$, where $[\cdot]$ denotes the binary notation of a number. Bob, given $S_1, S_2, \ldots, S_m$ computes $S'_i = \{\ (\ [i-1] \cdot u,\ [n-i] \cdot v\ )\ |\ (u, v) \in S_i\ \}$ and $S' = \cup_{i=1}^m S_i$. Since $[1, x'] \times [1, y'] \cap S' = \emptyset$ iff $[1, x] \times [1, y] \cap S_i = \emptyset$, they get the correct result by simulating the $ERQ_{n,l}$ protocol.

We are now ready for the main part of our proof. Given a protocol for $ERQ_{n,l}$, we use the first reduction above to get a $[t, a, b]^A$-protocol for

$$P_{100a}(ERQ_{\frac{n}{100a},\ l})$$

. We use the round elimination lemma to get a $[t-1, 120a, 120b]^B$-protocol for

$$ERQ_{\frac{n}{100a},\ l}.$$

The second reduction above gives us a $[t-1, 120a, 120b]^B$-protocol for

$$P^{12000b}(ERQ_{\frac{n}{100a}-\log(12000b),\ l/(12000b)}).$$

Using the round elimination lemma again, we get a $[t-2, 120^2a, 120^2b]^A$-protocol for

$$ERQ_{\frac{n}{100a}-\log(12000b),\ l/(12000b)}.$$

By doing this two round elimination repeatedly, and combining with the fact that there is clearly no $[0, a', b']$- protocol for $ERQ_{n^{\Omega(1)},l^{\Omega(1)}}$ for any a', b', we are done.

□

We can also use our technique to derive $\Omega(\sqrt{\log n})$ lower bounds for the one-dimensional counting and modulo-counting problems, and, in fact, for all the problems considered in [Mil94]. The proofs are similar to the above and are omitted from this extended abstract.

4.3 The "Greater Than" Problem

The GT_n function is defined as follows: Alice and Bob each gets an n-bit integer, x and y, resp., and they must decide whether $x > y$. It is easy to see that the deterministic communication complexity of GT_n is linear, and it is known that the randomized complexity is $O(\log n)$ [Ni93]. The upper bound requires $O(\log n)$ rounds of communication, and it is not hard to obtain a k-round protocol using $O(n^{1/k} \log n)$ bits of communication. Yao, in an unpublished result, shows that this is close to optimal. We can easily rederive his lower bound (in a somewhat weaker form) from the round elimination lemma.

Theorem 14 *There does not exist a randomized* $[k, n^{1/k}/120^k, n^{1/k}/120^k]$ *protocol for* GT_n*.*

Proof: The proof is by induction on k. We will show that a $[k, n^{1/k}/120^k, n^{1/k}/120^k]$ protocol for GT_n implies a similar one for $P_{n^{1/k}}(GT_{n'})$, for $n' = n^{(k-1)/k}$. Using the round elimination lemma this implies a $[k-1, n^{1/k}/120^{k-1}, n^{1/k}/120^{k-1}]$ protocol for $GT_{n'}$. A contradiction to the induction hypothesis is obtained since $n^{1/k} = n'^{1/(k-1)}$.

Here is the required reduction: To solve $P_{n^{1/k}}(GT_{n'})$ using a protocol for GT_n, Alice constructs an n-bit integer $\hat{x}$, by concatenating $x_1, ..., x_m$. Bob constructs an n-bit integer $\hat{y}$ by concatenating $x_1, ..., x_{i-1}, y$ and another $(n^{1/k} - i)n'$ one bits. One can easily verify that $\hat{x} > \hat{y}$ iff $x_i > y$.

□

4.4 Depth Hierarchy for Monotone AC^0

Let ${T_n}^k$ be the boolean function on n^k variables defined inductively as follows: ${T_n}^0(x) = x$, for odd k, ${T_n}^k$ is the OR of n copies of ${T_n}^{k-1}$, and for even k, ${T_n}^k$ is the AND of n copies of ${T_n}^{k-1}$. Each of the copies is a disjoint set of variables. Thus ${T_n}^k$ is defined by an AND/OR tree of fanin n and depth k.

It is clear that ${T_n}^k$ can be computed by a monotone depth k formula of size $N = n^k$, with the bottom gates being OR gates. In [KPPY84] it is proved that monotone depth k circuits with bottom gates being AND gates require exponential size to compute ${T_n}^k$. This lower bound is equivalent to a lower bound in communication complexity using the equivalence due to [KW90], (see also [NW93]). Our lemma allows us to re-derive this lower bound (in a somewhat weaker form).

Theorem 15 *[KPPY84] Any monotone depth k formula with bottom gates being AND gates requires size* $\Omega(n/120^k) = \Omega(N^{1/k}/120^k)$ *size to compute* ${T_n}^k$*.*

Comment: An exponential lower bound for depth k circuits directly follows by the straight forward simulation of depth k circuits by depth k formulae.
Proof: Let ${f_n}^k$ be the communication problem associated with the monotone formula complexity of ${T_n}^k$ ([KW90], see also [NW93]). (Here Alice is the AND player – holding a maxterm of ${T_n}^k$.) We will prove by induction on k that ${f_n}^k$ does not have $[k, n/120^k, n/120^k]^A$ protocols (we assume k is even, the odd case is simply dual). This clearly suffices to prove the theorem.

Inspection of ${f_n}^k$ reveals that it is completely equivalent to $P_n({f_n}^{k-1})$, only that Bob does not also get copies of the first $i - 1$ strings of Alice. Using the round elimination lemma we see that a $[k, n/120^k, n/120^k]^A$ protocol for ${f_n}^k$ implies a $[k-1, n/120^{k-1}, n/120^{k-1}]^B$ protocol for ${f_n}^{k-1}$, which by induction does not exist.

□

References

[Ajt88] M. Ajtai. A lower bound for finding predecessors in Yao's cell probe model. *Combinatorica*, 8:235–247, 1988.

[BF94] P. Beame, F. Fich, personal communication.

[DGS84] P. Duris, Z. Galil, G. Schnitger. Lower Bounds of Communication Complexity. In *Proc. 16th ACM Symposium on Theory of Computing (STOC)* (1984) 81-91.

[FKS84] M.L. Fredman, J. Komlòs, and E. Szemerédi. Storing a sparse table with O(1) worst case access time. *J. Ass. Comp. Mach.*, 31:538–544, 1984.

[HR88] B. Halstenberg, R. Reischuk: On Different Modes of Communication. In *Proc. 20th ACM Symposium on Theory of Computing (STOC)* (1988) 162-172.

[KW90] M. Karchmer and A. Wigderson. Monotone circuits for connectivity require super-logarithmic depth. *SIAM Journal on Discrete Mathematics*, 3, 1990.

[KPPY84] M. Klawe, W.J. Paul, N. Pippenger, M. Yannakakis: On Monotone Formulae with Restricted Depth In *Proc. 16th ACM Symposium on Theory of Computing (STOC)* (1984) 480–487.

[MS82] K. Mehlhorn and E. M. Schmidt. Las Vegas is better than determinism in VLSI and distributed computing. In *Proc. 14th ACM Symposium on Theory of Computing (STOC)* (1982) 330–337.

[Mil93] P.B. Miltersen. The bit probe complexity measure revisited. In *Proc. 10th Symp. on Theoretical Aspects of Computer Science (STACS)* (1993) 662–671.

[Mil94] P.B. Miltersen. Lower bounds for union-split-find related problems on random access machines. In *Proc. 26th ACM Symposium on Theory of Computing (STOC)*, (1994) 625–634.

[Mil95] P.B. Miltersen. On the cell probe complexity of polynomial evaluation. *Theoretical Computer Science*, to appear.

[Ni93] N. Nisan. The communication complexity of threshold gates. In *proc. of "Combinatorics, Paul Erdos is Eighty*, (1993) 301–315.

[NW93] N. Nisan and A. Wigderson. Rounds in Communication Complexity revisited. *SIAM J. Comp.*, 22:1, 211–219, 1993.

[Weg87] I. Wegener, *The Complexity of Boolean Functions*, Wiley-Teubner series in Computer Science, 1987.

[Wil83] D.E. Willard. Log-logarithmic worst case range queries are possible in space $\theta(n)$. *Inform. Process. Lett.*, 17:81–84, 1983.

[Xia92] B. Xiao. *New bounds in cell probe model.* PhD thesis, UC San Diego, 1992.

[Yao77] A.C. Yao. Probabilistic computations: Toward a unified measure of complexity. In *Proc. 18th IEEE Symposium on Foundations of Computer Science (FOCS)* (1977) 222–227.

[Yao79] A.C. Yao. Some complexity questions related to distributive computing. In *Proc. 11th ACM Symposium on Theory of Computing (STOC)*, (1979) 209–213.

[Yao81] A.C. Yao. Should tables be sorted? *J. Ass. Comp. Mach.*, 28:615–628, 1981.

[Yao83] A.C. Yao. Lower bounds by probabilistic arguments. In *Proc. 24th IEEE Symposium on Foundations of Computer Science (FOCS)* (1983) 420–428.

Appendix

Proof of the Richness Lemma

Proof: We first show the following, slightly stronger statement for deterministic protocols:

- Let f be a (u, v)-rich problem. If f has a deterministic $[a, b]$-protocol, then f contains a submatrix of dimensions at least $u/2^a \times v/2^{a+b}$ containing only 1-entries.

The proof is by induction in $a + b$. If $a + b = 0$, no communication takes place, so f must constant, and, since it is (u, v)-rich, we must have $|A| \geq u$, $|B| \geq v$ and $f(x, y) = 1$ for all x, y.

For the induction step, assume first that Alice sends the first bit in the protocol. Let A_0 be the inputs for which she sends 0, and A_1 be the inputs for which she sends 1. Let f_0 be the restriction of f to $A_0 \times B$ and let f_1 be the restriction of f to $A_1 \times B$. By a simple averaging argument either f_0 or f_1 is $(u/2, v/2)$-rich. Assume WLOG that it is f_0. Now, f_0 has an $[a-1, b]$-protocol, so by the induction hypothesis, f_0 contains a 1-matrix of dimensions at least $(u/2)/2^{a-1} \times (v/2)/2^{a-1+b}$ which is what we are looking for.

Assume next that Bob sends the first bit, at let B_0, B_1, f_0, f_1 be defined analogously. Either f_0 or f_1 is $(u, v/2)$ rich so either f_0 or f_1 contains by the induction hypothesis a 1-matrix of dimensions $u/2^a \times (v/2)/2^{a+b-1}$ which is what we are looking for. This completes the induction.

Now assume a randomized one-sided error protocol for f is given. By fixing the random coin tosses made by the protocol, we can convert it into a deterministic protocol computing a function f' with the following properties:

- $f(x, y) = 0 \Rightarrow f'(x, y) = 0$
- f' is $(u/4, v/4)$-rich.

By applying the deterministic version of the lemma to f', we are done.

□

Proof of the Round Elimination Lemma

Proof: Let $m = 100a$ and let $I = \{1, \ldots, m\}$.

Assume a randomzied protocol for $P_m(f)$ with error probability 1/3. By repeating it 120 times in parallel, and taking majority of the results, we get the error probability down to less than 1/4000.

For any distribution D on $X \times Y$ we will construct a deterministic $t - 1$ round algorithm for f that errs on at most 15% of the inputs weighted according to the distribution D. A randomized algorithm for f follows from Yao's version of the von Neuman minmax theorem [Yao77].

Define a distribution D^* on $X^m \times I \times Y$ as follows: For each $1 \leq j \leq m$ we choose (independently) (x_j, y_j) according to distribution D, and we choose i uniformly at random in I. We set $y = y_i$ (and throw away all other y_j's).

Let A be a deterministic algorithm for $P_m(f)$ that errs on a fraction of at most $1/4000$ of the input weighted by distribution D^* (such an algorithm exists by the easy direction of the minmax theorem).

Define S to be the set of $(\langle x_1, \ldots, x_m \rangle, i)$ for which

$$\Pr_{D*}[A \text{ errs} \mid \langle x_1, \ldots, x_m \rangle, i] \leq 1/20.$$

Consider the set R of $\mathbf{x} = \langle x_1, \ldots, x_m \rangle$ for which $(\mathbf{x}, i) \in S$ for at least $99/100$ of the possible values of i. Using the Markov inequality we see that $\Pr_{D^m}(R) \geq \frac{1}{2}$.

Since Alice sends a bits in her first message, she partitions R into at most 2^a sets, let T be the subset of R that has maximum weight, its weight is at least $\Pr_{D^m}(T) \geq \frac{\Pr_{D^m}(R)}{2^a} \geq 1/2^{a+1}$.

We now claim

- There exists $i \in I$, $a_1, a_2, \ldots, a_{i-1} \in X$, and a set $G \subseteq X$ with the following properties,
  1. $\Pr_D(G) \geq 0.9$
  2. For any $x \in G$, we can find $x_{i+1}, x_{i+2}, \ldots, x_m$, so that
  $$\langle a_1, \ldots, a_{i-1}, x, x_{i+1}, \ldots, x_m \rangle \in T$$
  and
  $$(\langle a_1, \ldots, a_{i-1}, x, x_{i+1}, \ldots, x_m \rangle, i) \in S.$$

Before we prove this claim, we show that it implies our lemma. Here is a $t-1$ round algorithm for f on inputs x and y:

- Alice, given x, constructs an input for A as follows: If $x \in G$ then she picks a sequence $\mathbf{x}$ that starts with with $a_1, \ldots, a_{i-1}, x$ such that $\mathbf{x} \in T$ and $(\mathbf{x}, i) \in S$. Such a sequence exists by the definition. If $x \notin G$ then she picks an arbitrary sequence.

- Bob, given y, constructs his input for A as follows: i is already defined, $x_j = a_j$ for all $j < i$, y is given to him.

- The two players run the algorithm A but skipping the first round of communication, instead assuming that the first message Alice sent was the one yielding T.

The probability that the algorithm errs when (x, y) are chosen according to D is given by $\Pr_D[\text{ error }] \leq \Pr_D[x \notin G] + \Pr_D[\text{ error } | x \in G]$. The first term is bounded from above by $\frac{1}{10}$, and to bound the second term we observe that for $x \in G$, the sequence $(\mathbf{x}, i)$ is in S, so the probability of error for a random y, given x is at most $\frac{1}{20}$. Thus the total probability of error is at most 0.15.

We now prove the claim, by showing that the procedure in Figure 1 is guaranteed to find i and $\langle a_1, a_2, \ldots, a_{i-1} \rangle$ with the correct properties. Assume that it fails. Note that by the definition of R (of which T is a subset), the first

```
$i := 1$
$T_1 := T$
do
    $T_i^1 := \{\mathbf{x} \in T_i | (\mathbf{x}, i) \in S\}$
    $T_i^0 := \{\mathbf{x} \in T_i | (\mathbf{x}, i) \notin S\}$
    if $\Pr(T_i^0 | T_i) \geq 0.05$ then
        Fix $a_i$ so that $\Pr(x_i = a_i | \mathbf{x} \in T_i^0)$ is maximized.
        $T_{i+1} := \{\mathbf{x} \in T_i^0 | x_i = a_i\}$
    elseif $Pr_D(x | \exists x_{i+1}, \ldots, x_n : (a_1, \ldots, a_{i-1}, x, x_{i+1}, \ldots, x_n) \in T_i^1) \geq 0.9$ then
        halt, $\{(a_1, \ldots, a_{i-1})\}$ is the sought after vector
    else
        Fix $a_i$ so that $\Pr(x_i = a_i | \mathbf{x} \in T_i^1)$ is maximized.
        $T_{i+1} := \{\mathbf{x} \in T_i^1 | x_i = a_i\}$
        $\{Pr_{D^{m-i}}(T_{i+1}) \geq Pr_{D^{m-i+1}}(T_i) * 0.95/0.9\}$
    endif
    $i := i + 1$
od
```

Figure 1: Procedure for constructing $\langle a_1, a_2, \ldots, a_i - 1 \rangle$

clause in the if-statement can be satisfied at most $m/100$ times, which means that

$$\Pr_D(T_m) \geq \Pr_{D^m}(T) \cdot (0.05)^{m/100} \cdot (0.95/0.9)^{99m/100}$$

$$\geq 2^{-(a+1)} (0.05^{1/100} \cdot (0.95/0.9)^{99/100})^{100a} \geq 2^{-(a+1)} 10^a > 1,$$

a contradiction.

□

Recent Publications in the BRICS Report Series

RS-94-41 Peter Bro Miltersen, Noam Nisan, Shmuel Safra, and Avi Wigderson. *On Data Structures and Asymmetric Communication Complexity*. December 1994. 17 pp.

RS-94-40 Luca Aceto and Anna Ingólfsdóttir. *CPO Models for GSOS Languages — Part I: Compact GSOS Languages*. December 1994. 70 pp. An extended abstract of the paper will appear in: *Proceedings of CAAP '95,* LNCS, 1995.

RS-94-39 Ivan Damgård, Oded Goldreich, and Avi Wigderson. *Hashing Functions can Simplify Zero-Knowledge Protocol Design (too)*. November 1994. 18 pp.

RS-94-38 Ivan B. Damgård and Lars Ramkilde Knudsen. *Enhancing the Strength of Conventional Cryptosystems*. November 1994. 12 pp.

RS-94-37 Jaap van Oosten. *Fibrations and Calculi of Fractions*. November 1994. 21 pp.

RS-94-36 Alexander A. Razborov. *On provably disjoint* NP*-pairs*. November 1994. 27 pp.

RS-94-35 Gerth Stølting Brodal. *Partially Persistent Data Structures of Bounded Degree with Constant Update Time*. November 1994. 24 pp.

RS-94-34 Henrik Reif Andersen, Colin Stirling, and Glynn Winskel. *A Compositional Proof System for the Modal* μ*-Calculus*. October 1994. 18 pp. Appears in: Proceedings of LICS '94, IEEE Computer Society Press.

RS-94-33 Vladimiro Sassone. *Strong Concatenable Processes: An Approach to the Category of Petri Net Computations*. October 1994. 40 pp.

RS-94-32 Alexander Aiken, Dexter Kozen, and Ed Wimmers. *Decidability of Systems of Set Constraints with Negative Constraints*. October 1994. 33 pp.

RS-94-31 Noam Nisan and Amnon Ta-Shma. *Symmetric Logspace is Closed Under Complement*. September 1994. 8 pp.