On Data Structures and Asymmetric Communication Complexity *

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.

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 $y \subseteq U$ of size

*A preliminary version of this paper appeared in the proceedings of the 27th ACM Symposium on Theory of Computing (STOC).

†BRICS, University of Aarhus. Supported by a postdoctoral fellowship from the Danish Natural Science Research Council through BRICS, University of Aarhus and by the ESPRIT II Basic Research Actions Program of the European Community under contract No. 7141 (project ALCOM II). Part of this work was done while visiting University of Toronto.

‡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. Supported by BSF 92-00106 and by a Wolfson research award, administered by the Israeli Academy of Sciences and Humanities. Part of this work was done while visiting BRICS at the University of Aarhus.

at most l, and the two players must decide if $x \in y$. 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 $f : Q \times D \to A$, where $f(x, y)$ is the answer to query x about data y. In the case of Boolean queries, we will have $A = \{0, 1\}$, but we will sometimes consider non-Boolean queries as well. A solution with parameters s, b and t, is a method of storing any $y \in D$ as a data structure $\phi(y)$ 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 *dictionary* problem where D is the set of subsets $y \subset \{0, \ldots, N-1\}$ of a certain size, Q is the set $\{0, \ldots, N-1\}$ and $f(x, y) = 1$ if and only if $x \in y$.

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 $x \in Q$, Bob gets $y \in D$, and they must determine $f(x, y)$. For the dictionary problem, the corresponding communication problem is thus $MEM_{N,l}$. If there is a 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 the number of bits $|x| = \log |Q|$ in the query is much smaller than the number of bits $|y| = \log |D|$ required to represent the stored data, 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 $y \in \{0,1\}^m$ using $s = m^{O(1)}$ cells each of size $b \geq \log m$ so that any query $x \in \{0,1\}^n$ can be answered in $t = O(n/(\log b - \log\log m))$ probes.

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 $f : X \times Y \to \{0,1\}$ 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 the matrix M with $M_{x,y} = f(x,y)$, 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.

We also present a version of the lemma applicable to two-sided error protocols. 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 $x \subseteq \{0,\ldots,N-1\}$ of size k, Bob gets $y \subseteq \{0,\ldots,N-1\}$ of size l, and they must decide if $x \cap y = \emptyset$. (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 $\{0, \ldots, N-1\}$, 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: Let $C = 99$ and $R = 4256$. Suppose there is a randomized $[t, a, b]^A$-protocol for solving $P_{Ra}(f)$. Then there is a randomized $[t-1, Ca, Cb]^B$-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. We demonstrate the power of the lemma by easily deriving several of the known lower bounds (though sometimes in a somewhat weaker form) and some new lower bounds, both for data structure problems and for other communication complexity problems. These include:

- Lower bounds for data structures for predecessor and parity prefix query problem in the cell probe model. Such bounds were first proved in [Ajt88, Xia92, Mil94, BF94].

- The first lower bound for the two-dimensional reporting range query problem in the cell probe model.

- 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.) Such a tradeoff was first proved by Smirnov [Smi88].

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 $x \in Q$, Bob gets $y \in D$, and they must determine $f(x,y)$.

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.*

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 the communication problem then the data structure problem has a solution with parameters $s = 2^{O(a)}$, $t = O(1)$, and b.*

Proof: Suppose a $[t, a, b]$-protocol is given with $t = O(1)$. For $y \in D$, define a data structure $\phi(y)$ representing y as follows. Let $v = (\alpha_1, \alpha_2, \ldots, \alpha_i)$, $i \leq t$ be a possible sequence of messages of Alice. For each such v, there is a cell $(\phi(S))_v$ in the data structure. The cell contains the message Bob would send after Alice's i'th message, given that his input is y, and Alice's i first messages are as described by v. Note that the number of cells in the data structure is $2^{O(at)} = 2^{O(a)}$. Given the data structure, we can answer a query x in time $O(t)$ by playing the role of Alice with input x and reading Bob's messages in the data structure.

□

An example of using Lemma 2 for constructing a data structure is given in Section 4.3.

A more general converse is, however, impossible. Using communication complexity, we can at most show an $\Omega(n/\log s)$ lower bound on the query time, since in this number of rounds, Alice can send her entire query to Bob when she sends $\log s$ bits in each round. However, there are well known data structure problems 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 $y \subseteq \{0,1\}^n$, so that for any $x \in \{0,1\}^n$, the query "$\exists z \in y \forall i : x_i \leq z_i$?" can be answered. No solution is known with worst case query time even polynomial in n when the structure size is polynomial, and it has been conjectured that no such structure exists [Riv76]. Yet not only does communication complexity fail to provide

bounds better than $n/\log s$, but for this problem, we only know how to show a $\sqrt{\log n}$ lower bound when s is polynomial in m, using the techniques of section 4. Getting bounds for this problem closer to $n/\log s$ using communication complexity is an interesting open problem.

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* $f : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}$ *the following holds with high probability: For any representation using* $s \leq 2^{n-1}/b$ *cells of size* b*, query time* $\Omega(m/b)$ *is necessary.*

Note that with twice as much storage, $2^n/b$ cells, the answer to every possible query could be stored and constant query time would be possible.

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

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 $O(n/(\log b - \log\log m))$.

Given a branching program for f of size $(n+m)^{O(1)} = m^{O(1)}$, and a data instance $y \in \{0,1\}^m$, eliminate all y_i-variables in the branching program, leaving only query variables x_i. The size has not increased. We store a pointer structure representing this new branching program. Since $b \geq \log m$, a pointer can be represented in a constant number of cells..

Given a query x, we simulate the stored branching program on x. 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. If $b = 2^r \log m$, we can in a constant number of cells represent a binary tree of depth r with pointers to branching program locations in the nodes and indices of x_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 $f : X \times Y \to \{0,1\}$, we identify f with the M with $M_{x,y} = f(x,y)$, 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.*

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 $|X| \geq u$, $|Y| \geq v$ 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 X_0 be the inputs for which she sends 0, and X_1 be the inputs for which she sends 1. Let f_0 be the restriction of f to $X_0 \times Y$ and let f_1 be the restriction of f to $X_1 \times Y$. 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, and let Y_0, Y_1, f_0, and 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.

□

Lemma 5 only shows lower bounds for one-sided error protocols. The following version of the lemma works for randomized protocols with two-sided error, but applies to a smaller range of problems. The lemma and its proof are very similar to a lemma for symmetric communication complexity by Yao [Yao83].

For finite sets S, T, the density of S in T is $|S \cap T|/|T|$. A communication problem $f : X \times Y \to \{0,1\}$ is α-dense if the density of $\{(x,y)|f(x,y) = 1\}$ in $X \times Y$ is at least α.

Lemma 6 *Let $\alpha, \epsilon > 0$. Let $f : X \times Y \to \{0,1\}$ be an α-dense problem. If f has a randomized (two-sided error) $[a,b]$-protocol, then there is a submatrix M of f of dimensions at least $|X|/2^{O(a)} \times |Y|/2^{O(a+b)}$ so that the density of 0-entries in M is at most ϵ (the constants in the big-O's depending on α and ϵ only).*

Proof: Given a randomized protocol, repeat it $O(1)$ times to get the error probability lower than δ where $\frac{2\delta}{\alpha-\delta} = \epsilon$.

By Yao's version of the von Neuman minmax theorem [Yao77], we can find a *deterministic* protocol with the same parameters which errs on a fraction of at most δ of $X \times Y$.

Let f' be the function computed by this protocol. The possible histories of the communication protocol induces a partition of the matrix $M_{f'}$ into disjoint submatrices. Consider the submatrices for which the two players answer 1. The union of these submatrices (i.e. $\{(x,y)|f'(x,y) = 1\}$) has density at least $\alpha - \delta$ in M_f. Furthermore, the set of pairs (x,y) for which $f(x,y) = 0$ has density at most $\frac{\delta}{\alpha-\delta}$ in $\{(x,y)|f'(x,y) = 1\}$. Consider the set S of 1-submatrices in which the density of such pairs is at most $\frac{2\delta}{\alpha-\delta} = \epsilon$. The union S' of these submatrices has density at least $\frac{\alpha-\delta}{2}$.

It follows that a fraction of at least $\frac{\alpha-\delta}{4}$ of the columns of f' each contains a fraction of at least $\frac{\alpha-\delta}{4}$ members of S'.

Now, by an induction similar to the one in Lemma 5, we can find a submatrix M of f', so that at least $\frac{\alpha-\delta}{4}|X|/2^{O(a)}$ columns of M contains at least $\frac{\alpha-\delta}{4}|Y|/2^{O(a+b)}$ entries from S'.

Moreover, the way the induction works implies that M *itself* is a matrix induced by a communication history, so it must be a matrix in S of dimensions at least $\frac{\alpha-\delta}{4}|X|/2^{O(a)} \times \frac{\alpha-\delta}{4}|Y|/2^{O(a+b)} \geq |X|/2^{O(a)} \times |Y|/2^{O(a+b)}$.

□

3.2 The membership problem

Let MEM_N be the *unrestricted* membership problem, where Alice gets $x \in \{0,1,\ldots,N-1\}$ and Bob gets some subset $y \subseteq \{0,1,\ldots,N-1\}$ with no restriction on the size of y and the two players must output 0 if $x \notin y$ and 1 if $x \in y$. Clearly, for any a, MEM_N (for N a power of two) has a deterministic $[a+1, N/2^a]$-protocol, where first Alice sends Bob the first a bits of her input x, then, for each $z \in \{0,1,\ldots,N-1\}$ with the same prefix as x, Bob tells Alice if $z \in y$, and finally Alice tells Bob the answer. We show that this is almost optimal, even for two sided error protocols.

Theorem 7 *For any randomized protocol $[a,b]$-protocol for MEM_N, $b \geq N/2^{O(a)}$*

Proof: If $a \geq \log N/2$, there is nothing to show, so assume $a < \log N/2$. Suppose we can find a submatrix $X_1 \times Y_1$ of MEM_N of dimensions at least $r \times s$ with a fraction of at most $\epsilon \leq \frac{1}{8}$ 0-entries. At least $s/2$ of the columns of $X_1 \times Y_1$ contains a fraction of at most 2ϵ 0-entries. Suppose a column contains exactly i zeroes. The set $y \subseteq \{0,1,2,\ldots,N-1\}$ corresponding to

this column is the disjoint union of a subset of X_1 of size $r-i$ and a subset of $\{0,1,2,\ldots,N-1\}-X_1$. Thus, we must have

$$s/2 \leq \sum_{i=0}^{2\epsilon r} \binom{r}{i} 2^{N-r} \leq 2^{N-\delta r}$$

where δ is a positive constant depending on ϵ.

Since MEM_N is $\frac{1}{2}$-dense, we have by Lemma 6, that we *can* find an $r \times s$ matrix with an ϵ-fraction 0-entries, where $r = N/2^{O(a)}$ and $s = 2^{N-O(a+b)}$. Combining, we get

$$2^{N-\delta(N/2^{O(a)})} = 2^{N-\delta r} \geq s/2 = 2^{N-O(a+b)}$$

so $a+b \geq N/2^{O(a)}$ and since $a < \log N/2$, we have $b \geq N/2^{O(a)}$.

□

We now consider the more complex problem $MEM_{N,l}$ where Alice gets $x \in \{0,1,\ldots,N-1\}$, Bob gets $y \subseteq \{0,1,\ldots,N-1\}$ of size at most l, and they must output 1 if $x \in y$, and 0 otherwise. Assume for convenience that N and l are powers of two, and that $l \leq N/2$. 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 8 *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 $a \geq \log l$, an $[2a, O(\log N + 2(a - \log l))]$-protocol.*
2. *For all $a \leq \log l$, a randomized one sided error $[O(a), O(l/2^a)]$-protocol.*

Proof: Deterministic Protocol: The protocol is based on perfect hashing [FKS84] and is thus an example of implicitly using Lemma 1 to give upper bound for communication problems rather than lower bounds for data structure problems.

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 his set y 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 his set y 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 $a \geq \log l$, if Alice has x and Bob has y, the two players do the following. Alice first sends Bob the first $2(a-\log l)$ bits of her input. She lets x' be the reminding $\log N - 2(a-\log l)$ bits of her input and Bob lets y' be the elements of y which has the prefix that Alice sends.

They then perform the $[2\log l, O(l(\log N - 2a - 2\log l)/2^{\log l})] = [2\log l, O(\log N - 2(a - \log l))]$ protocol just described on x' and y'.

Randomized Protocol: This is just a special case of the randomized protocol for Theorem 10).

□

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

Theorem 9 *If $MEM_{N,l}$ has a one-sided error $[a,b]$-protocol, with $a\log l$ then $2^a(a+b) = \Omega(l(\log N - \log l))$. If its negation has a one-sided error $[a,b]$-protocol, then $2^a(a+b) = \Omega(l)$, provided $l \leq N/2$.*

Proof: Assume without loss of generality that N and l are powers of two and that $a < \log l - 2$.

The $MEM_{N,l}$-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

$$\begin{array}{l} \binom{N-l/2^{a+2}}{l-l/2^{a+2}} \geq \binom{N}{l}/2^{a+b+2} \quad \Rightarrow \\ 2^{a+b+2} \geq \binom{N}{l}/\binom{N-l/2^{a+2}}{l-l/2^{a+2}} \end{array}$$

Note that for any integer t, we have

$$\binom{N}{l}/\binom{N-t}{l-t} = \frac{N(N-1)\cdots(N-t+1)}{l(l-1)\cdots(l-t+1)}$$

and since $\frac{N}{l} < \frac{N-1}{l-1} < \cdots < \frac{N-t+1}{l-t+1}$, we have

$$\binom{N}{l}/\binom{N-t}{l-t} > (\frac{N}{l})^t$$

Thus, continuing the implications above, we get

$$\begin{array}{rl} 2^{a+b+2} \geq (N/l)^{l/2^{a+2}} & \Rightarrow \\ \log 2^{a+b+2} \geq \log((N/l)^{l/2^{a+2}}) & \Rightarrow \\ (a+b+2) \geq l/2^{a+2}\log(N/l) & \Rightarrow \\ 2^{a+2}(a+b+2) \geq l(\log N - \log l) & \end{array}$$

as desired.

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 $(N-l)/2^{a+2} \times \binom{N}{l}/2^{a+b+2}$. But if the non-membership matrix contains a 1-submatrix of dimensions $r \times s$, then $\binom{N-r}{l} \geq s$, so

$$\begin{array}{l} \binom{N-\frac{N-l}{2^{a+2}}}{l} \geq \binom{N}{l}/2^{a+b+2} \quad \Rightarrow \\ 2^{a+b+2} \geq \binom{N}{l}/\binom{N-\frac{N-l}{2^{a+2}}}{l} \end{array}$$

Note that for any integer t, we have

$$\binom{N}{l} / \binom{N-t}{l} = \frac{N(N-1)\cdots(N-l+1)}{(N-t)(N-t-1)\cdots(N-t-l+1)}$$

and since $\frac{N}{N-t} < \frac{N-1}{N-t-1} < \cdots < \frac{N-l+1}{N-t-l+1}$, we have

$$\binom{N}{l} / \binom{N-t}{l} > (\frac{N}{N-t})^l.$$

Thus, continuing the implications above, we get

$$\begin{array}{rl} 2^{a+b+2} \geq (\frac{N}{N-\frac{N-l}{2^{a+2}}})^l & \Rightarrow \\ a+b+2 \geq l \log(\frac{N}{N-\frac{N-l}{2^{a+2}}}) & \Rightarrow \\ a+b+2 \geq l \log(1+\frac{1}{2^{a+2}}) & \Rightarrow \\ 2^a(a+b) = \Omega(l) & \end{array}$$

□

The deterministic upper bounds and the lower bounds for one-side error protocols are tight in the following sense: There are constants $c, c' > 0$ so that for $l \leq N^{1-\epsilon}$ and $a \leq \log l$, $b = l \log N/2^{ca}$ is sufficient and $b = l \log N/2^{c'a}$ is not sufficient. The bounds for randomized one-sided error protocols for non-membership are tight in a stronger sense: There are constants $c, c' > 0$, so that for any $l \leq n/2$ and a, $b = l/2^{ca}$ is sufficient and $b = l/2^{c'a}$ is not sufficient.

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 $x \subseteq \{0, \ldots, N-1\}$ of size k, Bob gets $y \subseteq \{0, \ldots, n-1\}$ of size l, and they decide if $x \cap y = \emptyset$. The symmetric version of this problem is, of course, well studied. We give an upper and a lower bound. As yet, the tradeoffs are not completely understood.

Theorem 10 *$DISJ_{N,k,l}, k < l < N/2$ 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 $[O(a), O(l \log(k/a))]$-protocol 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 $\{0, \ldots, N-1\}$. Each round Alice will send to Bob the next i such that $x \subseteq R_i$, Bob will update his set $y \leftarrow y \cap R_i$, and will send to Alice $j - i$ for the next j such that $y \subseteq R_j$ (the new y), and then Alice will update $x \leftarrow x \cap R_j$. If at any point during the protocol x or y 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 x (resp. y). If x and y are disjoint then the expected size of both x and y 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

x and y do not become empty after so many bits have been sent then, with high probability, x and y were not disjoint to begin with.

If $a \geq k$ then Alice starts by sending Bob the first a/k indices i for which $x \subseteq R_i$. This allows Bob to reduce the size of hist set y (assuming that it is disjoint from x) 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 for which $y \subseteq R_i$, reducing the size of x to $O(a)$ with high probability, assuming the sets were disjoint.

□

Theorem 11 *Let $k \leq l < N/2$. 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 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 $x_1, x_2, \ldots, x_r$ and let the columns be indexed by the sets $y_1, y_2, \ldots, y_s$. We then have that $x_i \cap y_j = \emptyset$ for all i, j. Consider the set $x = \cup_j x_j$. Since every x_j is a subset of x of size k and there are least $\binom{N}{l}/2^{a+b}$ different x_j's, we must have $\binom{|x|}{k} \geq \binom{N}{l}/2^{a+b}$. Let $t = (N-l)/2^{a/k}$. Since $\binom{t}{k} < \binom{N-l}{k}/2^a$, we must have $|x| = |\cup x_i| > t$ and therefore $|\cup y_i| < N - t$. Thus, we must have $\binom{N}{l}/2^{a+b} > \binom{N-t}{l}$ and therefore $2^{a+b} > \binom{N}{l}/\binom{N-t}{l} > (\frac{N}{N-t})^l > (1+t/N)^l = (1+(N-l)/2^{a/k}N)^l \geq (1+2^{-a/k-1})^l$. The conclusion follows.

□

It would be interesting to give good lower bounds for two sided error protocols solving disjointness.

3.4 The Span Problem

The membership and disjointness problems exhibit 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 *SPAN*, 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 $x \in y$.

Theorem 12 *In any $[a, b]$ one-sided error randomized protocol for SPAN either $a = \Omega(n)$ or $b = \Omega(n^2)$.*

Proof: For the proof let us assume that y is of dimension exactly $n/2$, and is given by its basis.

Using Lemma 5, it suffices to show

1. *SPAN* is $(2^{n/2}, 2^{n^2/4})$-rich, and
2. *SPAN* does not contain a 1-monochromatic submatrix of dimensions $2^{n/3} \times 2^{n^2/6}$.

For 1, notice that every subspace of Z_2^n of dimension exactly $n/2$ contains exactly $2^{n/2}$ vectors, and that there are greater than $2^{n^2/4}$ subspaces of dimension $n/2$. To see this, we count the number of ways to choose a basis for such a space (i.e., to choose $n/2$ independent vectors). There are $2^n - 1$ possibilities to choose the first basis element (different from $\vec{0}$), $2^n - 2$ to choose the second, $2^n - 4$ to choose the third etc. Also note that each basis is chosen this way $\frac{n}{2}!$ times. Hence the number of bases is $\prod_{i=0}^{n/2-1}(2^n - 2^i)/\frac{n}{2}!$. Now, each subspace has a lot of bases. By a similar argument, their number is $\prod_{i=0}^{n/2-1}(2^{n/2} - 2^i)/\frac{n}{2}!$. Hence the total number of subspaces is:

$$\frac{\prod_{i=0}^{n/2-1}(2^n - 2^i)}{\prod_{i=0}^{n/2-1}(2^{n/2} - 2^i)} = \prod_{i=0}^{n/2-1} \frac{2^n - 2^i}{2^{n/2} - 2^i} \geq \prod_{i=0}^{n/2-1} 2^{n/2} = 2^{n^2/4}.$$

For 2, consider a 1-rectangle with at least $2^{n/3}$ rows. Note that any $2^{n/3}$ vectors span a subspace of Z_2^n of dimension $n/3$ and that, by a similar argument to the one presented above, the number of subspaces of dimension $n/2$ that contain a given subspace of dimension $n/3$ is

$$\frac{\prod_{i=0}^{n/6-1}(2^n - 2^{n/3+i})}{\prod_{i=0}^{n/6-1}(2^{n/2} - 2^{n/3+i})} = \prod_{i=0}^{n/6-1} \frac{2^n - 2^{n/3+i}}{2^{n/2} - 2^{n/3+i}} \leq \prod_{i=0}^{n/6-1} 2^n = 2^{n^2/6},$$

as needed.

□

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 13 (Round elimination lemma) *Let $\epsilon, \delta > 0$ be given so that $\delta \leq \frac{1}{100}\epsilon^2(-\ln\frac{\epsilon}{8})^{-1}$ and let $m \geq 20(a \ln 2 + \ln 5)\epsilon^{-1}$. Suppose there is a randomized $[t, a, b]^A$-protocol with error probability δ for solving $P_m(f)$. Then there is a randomized $[t-1, a, b]^B$-protocol with error probability ϵ for solving f.*

Remarks:

1. The above lemma is interesting even in the case where Bob does not get copies of $x_1, ..., x_{i-1}$; we need the stronger version as stated for our purposes.
2. Is the increase in error probability necessary? With a smaller increase, some of the lower bounds which follow from the lemma would be improved.
3. The lemma applies to randomized two-sided error computation. It would be interesting to get a similar theorem for deterministic computation.

Proof: Assume a randomized protocol for $P_m(f)$ with error probability δ. For any distribution D on $X \times Y$ we will construct a deterministic $t-1$ round algorithm for f that errs on at most $\epsilon/2$ of the inputs weighted according to the distribution D. A randomized algorithm for f with error probability ϵ follows from Yao's version of the von Neuman minmax theorem [Yao77].

Let $I = \{1, \ldots, m\}$. 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 δ 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

$$\Pr_{D*}[A \text{ errs} \mid \langle x_1, \ldots, x_m \rangle, i] \leq \frac{\epsilon}{4}.$$

Consider the set R of $\mathbf{x} = \langle x_1, \ldots, x_m \rangle$ for which $(\mathbf{x}, i) \in S$ for at least $1 - 5\delta\epsilon^{-1}$ of the possible values of i. Using the Markov inequality we see that $\Pr_{D^m}(\bar{R}) \leq \frac{4}{5}$ and hence $\Pr_{D^m}(R) \geq \frac{1}{5}$.

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 $\Pr_{D^m}(T) \geq \frac{\Pr_{D^m}(R)}{2^a} \geq \frac{1}{5 \cdot 2^a}$.

We now claim

- There exists $i \in I$, $q_1, q_2, \ldots, q_{i-1} \in X$, and a set $G \subseteq X$ with the following properties,
  1. $\Pr_D(G) \geq \frac{\epsilon}{4}$
  2. For any $x \in G$, we can find $x_{i+1}, x_{i+2}, \ldots, x_m$, so that $\langle q_1, \ldots, q_{i-1}, x, x_{i+1}, \ldots, x_m \rangle \in T$ and $(\langle q_1, \ldots, q_{i-1}, x, x_{i+1}, \ldots, x_m \rangle, i) \in S$.

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 $q_1, \ldots, q_{i-1}, x$ 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, $x_j = q_j$ 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 $\frac{\epsilon}{4}$, 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 $\frac{\epsilon}{4}$. Thus the total probability of error is at most $\epsilon/2$, as desired.

$i := 1$
$T_1 := T$
`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` $\Pr(T_i^0 | T_i) \geq \frac{\epsilon}{8}$ `then`
    Fix q_i so that $\Pr(x_i = q_i | \mathbf{x} \in T_i^0)$ is maximized.
    $T_{i+1} := \{\mathbf{x} \in T_i^0 | x_i = q_i\}$
  `elseif` $Pr_D(x | \exists x_{i+1}, \ldots, x_n : (q_1, \ldots, q_{i-1}, x, x_{i+1}, \ldots, x_n) \in T_i^1) \geq \frac{\epsilon}{4}$ `then`
    `halt`, $\{(q_1, \ldots, q_{i-1})\}$ is the sought after vector
  `else`
    Fix q_i so that $\Pr(x_i = q_i | \mathbf{x} \in T_i^1)$ is maximized.
    $T_{i+1} := \{\mathbf{x} \in T_i^1 | x_i = q_i\}$
    $\{Pr_{D^{m-i}}(T_{i+1}) \geq Pr_{D^{m-i+1}}(T_i) * \frac{1-\frac{\epsilon}{8}}{1-\frac{\epsilon}{4}}\}$
  `endif`
  $i := i + 1$
`od`

Figure 1: Procedure for finding i and $\langle q_1, q_2, \ldots, q_{i-1} \rangle$

We now prove the claim, by showing that the procedure in Figure 1 is guaranteed to find i and $\langle q_1, q_2, \ldots, q_{i-1} \rangle$ with the correct properties. Assume to the contrary that it fails.

In the i'th iteration the procedure has found $\mathbf{q} = (q_1, q_2, \ldots, q_{i-1})$ and a subset T_i of X^{m-i+1} with the property that $\mathbf{q} \cdot T_i \subseteq T$. It is easily checked that for all values of i, T_i has positive density, i.e. it is not empty. We now show that the first clause in the if-statement can be satisfied in at most $(5\delta\epsilon^{-1})m$ iterations. Suppose after i iterations it has been satisfied in more. Then for all values $\mathbf{x}$ in $\mathbf{q} \cdot T_i$, for at least a $5\delta\epsilon^{-1}$ fraction of the values in I, $(\mathbf{x}, i)$ is not in S. By the definition of R of which T and hence $\mathbf{q} \cdot T_i$ is a subset, this means that T_i is in fact empty, a contradiction. This means that

$$\begin{aligned}
\Pr_D(T_m) &\geq \Pr_{D^m}(T) \cdot (\frac{\epsilon}{8})^{(5\delta\epsilon^{-1})m} \cdot (\frac{1-\frac{\epsilon}{8}}{1-\frac{\epsilon}{4}})^{(1-5\delta\epsilon^{-1})m} \\
&\geq \frac{1}{5 \cdot 2^a} [(\frac{\epsilon}{8})^{5\delta\epsilon^{-1}} (1 + \frac{\epsilon}{8})^{1-5\delta\epsilon^{-1}}]^m \\
&\geq e^{-(\ln 2)a - \ln 5 + [5\delta\epsilon^{-1} \ln(\frac{\epsilon}{8}) + (1-5\delta\epsilon^{-1}) \ln(1+\frac{\epsilon}{8})]m} \\
&\geq e^{-(\ln 2)a - \ln 5 + [-\frac{\epsilon}{20} + \frac{\epsilon}{9}]m} \\
&> 1,
\end{aligned}$$

a contradiction.

□

For our applications, it is convenient to have a version of the round elimination lemma with a fixed error probability of 1/3.

Lemma 14 (Round elimination lemma, fixed error probability) *Let $C = 99$ and $R = 4256$. Suppose there is a randomized $[t, a, b]^A$-protocol with error probability $1/3$ for solving $P_{Ra}(f)$. Then there is a randomized $[t, Ca, Cb]^B$-protocol with error probability $1/3$ for solving f.*

Remark: Intuitively, the values of C and R seem much higher than necessary. Indeed, we don't know if $R = 2$ and $C = 1$ is sufficient. If so, some of the lower bounds which follow could be improved. A counterexample showing that $R = 2$ and $C = 1$ does not yield a valid statement for either randomized or deterministic protocols would also be of interest.

Proof: Repeat the protocol 99 times in parallel and take majority of the results. It is easily checked that this reduces the error probability to less than $\frac{1}{100}(1/3)^2(-\ln(1/24))^{-1}$. Now apply Lemma 13 on the repeated protocol.

□

4.2 Predecessor query problems

Ajtai [Ajt88] gave a lower bound for the problem of storing a subset $y \subseteq U = \{0, \ldots, 2^n - 1\}$ so that for any $x \in U$, the predecessor query "What is $\max\{z \in y | z \leq x\}$" can be answered efficiently. His proof is quite complicated. We reprove his lower bound quite easily using the round elimination lemma.

In fact, we show the lower bound for the *prefix parity* problem of storing a subset $y \subseteq U$ so that for any $x \in U$, the query "What is $|\{z \in y | z \leq x\}| \bmod 2$?" can be answered efficiently. It is not difficult to see that this problem reduces to the predecessor problem: Given a solution to the predecessor problem, we simply combine it with a perfect hash table containing, for each element z in the set y to be stored, z's rank in y. Thus, lower bounds for the prefix parity problem also hold for the predecessor problem.

By Lemma 1, we should consider the communication problem $PAR_{n,l}$ where Alice gets $x \in U$, Bob gets $y \subseteq U$ of size at most l and the players must determine $|\{z \in y | z \leq x\}| \bmod 2 = |y \cap [0, x]| \bmod 2$.

Theorem 15 *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 $PAR_{n,l}$ 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 PAR to $P_m(PAR)$ or $P^m(PAR)$, eliminating one round. We also need to reduce instances of $P_m(PAR)$ or $P^m(PAR)$ to PAR. The following two reductions take care of that:

Suppose that m divides n. A communication protocol for $PAR_{n,l}$ can be used as a protocol for $P_m(PAR_{n/m,l})$ as follows: Alice, given $x_1, \ldots, x_m$, computes the concatenation $\hat{x} = x_1 \cdot x_2 \cdot \cdots \cdot x_m$. Bob, given y, i, and $x_1, \ldots, x_{i-1}$, computes $\hat{y} = \{(x_1 \cdot x_2 \cdot \cdots \cdot x_{i-1} \cdot u \cdot 0^{n - \frac{in}{m}} | u \in y\}$. Since $|\hat{y} \cap [0, \hat{x}]| = |y \cap [0, x_i]|$, they get the correct result by simulating the $PAR_{n,l}$ protocol on inputs $\hat{x}, \hat{y}$.

Suppose m is a power of two which divides l. A communication protocol for $PAR_{n,l}$ can be used as a protocol for $P^m(PAR_{n-\log m-1,l/m})$ as follows: Alice, given a $n-\log m+1$ bit string x and i, computes $x' = [i-1]\cdot 0\cdot x$, where $[i-1]$ denotes the binary notation of $i-1$ (which contains $\log m$ bits). Bob, given $y_1, y_2, \ldots, y_m$, where each y_j is a subset of the $n-\log m-1$ bit strings, first embeds each y_j in the set of $n-\log m$ bit strings by prefixing the elements by a 0. Then, for each j, he adds the string $1^{n-\log m}$ to the set y_j if it has an odd number of elements. This ensures that the total number of elements in each y_j is even. Then he computes $\hat{y}_j = \{\ [j-1]\cdot u,\ \mid\ u \in y_j\ \}$ and $\hat{y} = \cup_{j=1}^{m}\hat{y}_j$. Since $|\hat{y}\cap[0,\hat{x}]| \equiv |y_i \cap [0,x]| \pmod 2$, they get the correct result by simulating the $PAR_{n,l}$ protocol on inputs $\hat{x}, \hat{y}$.

We are now ready for the main part of our proof. Given a protocol for $PAR_{n,l}$, we use the first reduction above to get a $[t,a,b]^A$-protocol for $P_{Ra}(PAR_{\lfloor\frac{n}{Ra}\rfloor,\ l})$. We use the round elimination lemma to get a $[t-1,Ca,Cb]^B$-protocol for

$$PAR_{\lfloor\frac{n}{Ra}\rfloor,\ l}.$$

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

$$P^{CRb}(PAR_{\lfloor\frac{n}{Ra}\rfloor-\lceil\log(CRb)\rceil-1,\ \lfloor\frac{l}{CRb}\rfloor}).$$

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

$$PAR_{\lfloor\frac{n}{Ra}\rfloor-\lceil\log(CRb)\rceil-1,\ \lfloor\frac{l}{CRb}\rfloor}.$$

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

□

Using Lemma 1, we get the lower bounds for the data structure problems as immediate corollaries.

Corollary 16 *In any solution to the prefix parity (and the predecessor) problem, if* $(n|y|)^{O(1)}$ *cells, each containing* $\log^{O(1)}|U|$ *bits are used to store the set* y*, query time is at least* $\Omega(\sqrt{\log\log|U|})$ *as a function of* $|U|$ *and at least* $\Omega(\log^{1/3}|y|)$ *as a function of* $|y|$*.*

The corresponding best known upper bounds are $O(\log\log|U|)$ using compressed van Emde Boas trees [Wil83] and $O(\log^{1/2}|y|)$ using fusion trees [FW93] or packed B-trees [And95].

Ajtai's paper contains the $\Omega(\sqrt{\log\log|U|}$ lower bound for the predecessor problem. Xiao [Xia92] and, independently, Beame and Fich [BF94] improved this to $\Omega(\log\log|U|/\log\log\log|U|)$. The round elimination lemma does not seem to be powerful enough to give this improved lower bound (but if the factor C in the lemma was replaced by 1, it would be).

4.3 Range query problems

In this section we analyze 1- and 2- dimensional *reporting* range query problems using communication complexity.

Let $U = \{0, \ldots, 2^n - 1\}$ and let $r \in \{1, 2\}$. The r-dimensional reporting range query problem is as follows: Given a data set $y \subseteq U^r$, construct a static data structure using at most s memory cells, each containing $b = O(n)$ bits, so that for any interval $x = [x_1, x_2]$ (for $r = 1$) or box $x = [x_1, x_2] \times [z_1, z_2]$ (for $r = 2$) we can answer the query "What is $x \cap y$?" efficiently.

We consider solutions with query time of the form $t = O(t_0 + k)$, where k is the number of points reported, i.e. $k = |x \cap y|$. We want to minimize t_0 while keeping s reasonably small.

We show two bounds: An $O(1)$ *upper* bound on t_0 for reporting queries in the one-dimensional case with $s = O(n|y|)$, and an $\Omega((\log |y|)^{1/3})$ *lower* bound on t_0 for 2-dimensional queries for $r > 1$ when $s = (n|y|)^{O(1)}$.

Previously, lower bound for reporting range queries were only given in *structured* models of computations, namely the *pointer machine* model [Cha90] and the *layered partition* model [AS95].

For the upper bound, we first consider the simpler problem, where we merely have to return some element of $x \cap y$ if one exists. We find it most convenient to express the upper bound in terms of communication complexity and use Lemma 2. This yields the space bound $s = O((n|y|)^{O(1)})$, which we will afterward optimize to $O(n|y|)$. We need a protocol for the communication problem $RQ_{n,l}$, where Alice gets an interval $[x_1, x_2]$, Bob gets a set $y \subseteq U$ of size at most l and the players must agree on an element in $[x_1, x_2] \cap y$ if one exists.

Theorem 17 *$RQ_{n,l}$ has an $[O(1), \log n + O(\log l), n]$-protocol.*

Proof: Identify x_1 and x_2 with their binary representation, and let $i \in \{0, \ldots, n-1\}$ be the most significant bit where x_1 and x_2 differ, and let w be their common prefix of length i. Since $x_1 < x_2$, we have $x_{1,(i+1)} = 0$ and $x_{2,(i+1)} = 1$. We can write $[x_1, x_2] = [x_1, z-1] \cup [z, x_2]$, where $z = w10^{n-i-1}$. Our protocol determines if $[x_1, x_2] \cap y \neq \emptyset$ and returns an element if it is not, by doing the same task on $[x_1, z-1] \cap y$ and $[z, x_2] \cap y$. We only describe the second part, the first is similar.

Alice sends i to Bob. They now determine if there an element in y starting with the prefix $w1$. This is done by running the deterministic $[O(\log l), O(\log n)]$-membership protocol (Theorem 8) with Alice's input being w and Bob's input being the set of i bit-prefixes of his set. If there isn't such an element $[z, x_2] \cap y$ is empty. Otherwise, the membership protocol also tells Bob exactly what w is, and he can send Alice the smallest element in y with prefix $w1$. Alice then checks if x_2 is smaller than this element, in which case $[z, x_2] \cap y$ is empty, otherwise the element sent to Alice by Bob is returned by the protocol. This completes the protocol.

□

Using Lemma 2, this yields a data structure using space $s = O((n|y|)^{O(1)})$ for storing a set y, and a constant time algorithm which on input $[x_1, x_2]$ returns a member of $y \cap [x_1, x_2]$, if such an element exist. By inspecting the data structure, we see that it really consists of $n + 1$

dictionaries, $D_0, D_1, \ldots, D_n$, with D_i associating a set of size $\leq |y|$ of strings of length i with elements of y. By implementing each dictionary using the linear space, constant query time solution of [FKS84], we get an $O(n|y|)$ space solution.

We now generalize this to reporting queries. We augment the data structure above with an ordered, doubly linked list containing all the elements of y and a linear space, constant time dictionary, associating for each element $x \in y$, a pointer to x's copy in the list. When a query $[x_1, x_2]$ we use the data structure above to find an element $x \in y \cap [x_1, x_2]$ if such an element exists. Using the dictionary, we then find x in the doubly linked list, and can now report all elements in $y \cap [x_1, x_2]$ in linear time by tracing the list in both directions.

Our lower bound shows that the 2-dimensional problem is more difficult than the 1-dimensional one:

Theorem 18 *In any solution to the 2-dimensional reporting range query problem with query time $t_0 + O(k)$, if $(n|y|)^{O(1)}$ cells, each containing $\log^{O(1)} |U|$ bits are used to store y, then $t_0 \geq \Omega(\log^{1/3} |y|)$.*

Proof: The proof is a reduction from the prefix parity problem. Given a subset $y = \{y_1, y_2, .., y_m\}$ of U, we want to encode it so we can answer queries "What is $|y \cap [1, x]|$ mod 2?" Assume without loss of generality that m is even.

Given a solution to the reporting range query problem, we construct the data structure corresponding to the following subset $\hat{y}$ of U^2:

$$\hat{y} = \{(y_{2j-1}, n - y_{2j} + 1) | 1 \leq j \leq m/2\}$$

Now, $|y \cap [0, i]|$ is odd if and only if $\hat{y} \cap ([0, i] \times [0, n - i])$ contains 1 element and $|y \cap [0, i]|$ is even if and only if $\hat{y} \cap ([0, i] \times [0, n - i])$ is empty. We can find out which is the case in time $t_0 + O(1)$. The lower bound now follows from Corollary 16.

□

The corresponding best known upper bound is $O(\sqrt{\log |y|})$ using fusion trees [FW93] or packed B-trees [And95].

4.4 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. Smirnov [Smi88] shows that this is close to optimal, and Yao (unpublished) improves Smirnov's bounds slightly. We can easily rederive the lower bound (in a somewhat weaker form) from the round elimination lemma.

Theorem 19 *Let $C = 99$. There does not exist a randomized $[k, n^{1/k} C^{-k}, n^{1/k} C^{-k}]$-protocol for GT_n.*

Proof: The proof is by induction on k. We will show that a $[k, n^{1/k}C^{-k}, n^{1/k}C^{-k}]$ 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 $[k-1, n^{1/k}C^{-(k-1)}, n^{1/k}C^{-(k-1)}]$ protocol for $GT_{n'}$. This is a contradiction to the induction hypothesis 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 $x_i > y$.

□

4.5 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 fan-in 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 20 *[KPPY84] Let $C = 99$. Any monotone depth k formula with bottom gates being AND gates requires size $\Omega(nC^{-k}) = \Omega(N^{1/k}C^{-k})$ 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 $[k, nC^{-k}, nC^{-k}]^A$ 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 $[k, nC^{-k}, nC^{-k}]^A$ protocol for ${f_n}^k$ implies a $[k-1, nC^{-(k-1)}, nC^{-(k-1)}]^B$ protocol for ${f_n}^{k-1}$, which by induction does not exist.

□

Acknowledgment

We would like to thank Eyal Kushilevitz for assistance on the writeup of Theorem 12.

References

[And95] A. Andersson. Sublogarithmic searching without multiplications. In *Proc. FOCS '95*, to appear.

[AS95] A. Andersson, K. Swanson. On the Difficulty of Range Searching. In *Proc. 4th International Workshop on Algorithms and Data Structures (WADS)* (1995) 473–481.

[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.

[Cha90] B. Chazelle. Lower bounds for orthogonal range searching, I: the reporting case. *J. Ass. Comp. Mach.*, 37:200–212, 1990.

[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.

[FW93] M.L. Fredman, D. Willard. Surpassing the information theoretic bound with fusion trees. *J. Comput. System Sci.*, 47:424–436, 1993.

[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 superlogarithmic depth. *SIAM Journal on Discrete Mathematics*, 3:255–265, 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*, 143:167–174, 1995.

[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.

[Riv76] R. Rivest. Partial-Match Retrieval Algorithms. *SIAM J. Comp.*, 5:19–50, 1976.

[Smi88] D.V. Smirnov, *Shannon's information methods for lower bounds for probabilistic communication complexity.* Master's Thesis, Moscow University, 1988.

[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.