Complexity classes in communication complexity theory
(preliminary version)

László Babai
Eötvös University, Budapest
and the University of Chicago

Péter Frankl
C. N. R. S., Paris

János Simon
University of Chicago

Abstract. We take a complexity theoretic view of A. C. Yao's theory of communication complexity. A rich structure of natural complexity classes is introduced. Besides providing a more structured approach to the complexity of a variety of concrete problems of interest to VLSI, the main objective is to exploit the analogy between Turing machine (TM) and communication complexity (CC) classes. The latter provide a more amicable environment for the study of questions analogous to the most notorious problems in TM complexity.

Implicitly, CC classes corresponding to P, NP, $coNP$, BPP and PP have previously been considered. Surprisingly, $P^{cc} = NP^{cc} \cap coNP^{cc}$ is known [AUY]. We develop the definitions of $PSPACE^{cc}$ and of the polynomial time hierarchy in CC. Notions of reducibility are introduced and a natural complete member in each class is found. $BPP^{cc} \subseteq \Sigma_2^{cc} \cap \Pi_2^{cc}$ [Si2] remains valid. We solve the question that $BPP^{cc} \supseteq NP^{cc}$ by proving an $\Omega(\sqrt{n})$ lower bound for the bounded-error complexity of the $coNP^{cc}$-complete problem "disjointness". Similar lower bounds follow for essentially any nontrivial monotone graph property. Another consequence is that the deterministically exponentially hard "equality" relation is not NP^{cc}-hard with respect to oracle-protocol reductions.

We prove that the *distributional complexity* of the disjointness problem is $O(\sqrt{n} \log n)$ under *any product measure* on $\{0,1\}^n \times \{0,1\}^n$. This points to the difficulty of improving the $\Omega(\sqrt{n})$ lower bound for the B2PP complexity of "disjointness".

The variety of counting and probabilistic classes appears to be greater than in the Turing machine versions. Many of the simplest graph problems (undirected reachability, planarity, bipartiteness, 2-CNF-satisfiability) turn out to be $PSPACE^{cc}$-hard.

The main open problem remains the separation of the hierarchy, more specifically, the conjecture that $\Sigma_2^{cc} \neq \Pi_2^{cc}$. Another major problem is to show that $PSPACE^{cc}$ and the probabilistic class UPP^{cc} are not comparable.

1. Introduction

Motivated by VLSI applications, research in communication complexity has so far mainly focused on lower bounds for protocols computing specific functions.

In this paper we take a look at communication complexity from the point of view of ("machine based") complexity theory. We find a rich structure of natural complexity classes, providing a structured framework for the classification of various concrete functions, by introducing notions of reducibility and highlighting complete problems in different classes. This stucture may occasionally serve as a guide to finding lower bounds of significance to VLSI, although this should not be the primary objective of this theory. An example is given in Corollary 9.6; the recognition that simple graph problems such as connectedness between a pair of points are $PSPACE^{cc}$-hard has lead to an $\Omega(n)$ lower bound for the bounded-error probabilistic complexity of these problems.

The primary goal, however, is to gain insight into the nature of alternation, counting and probabilistic complexity in a context where the chances of progress might be greater than for the analogous questions in Turing machine complexity.

In the basic model, introduced by Yao [Ya1], two communicating parties (North and South) want to cooperatively determine the value $f(x,y)$ of a Boolean function f in $2n$ variables. Both North and South have complete information about f and unlimited computational power but receive only half of the input (x and y, resp.) ($x, y \in \{0,1\}^n$). They exchange bits according to some protocol until one of them (South) declares the value of $f(x,y)$. The objective is to minimize the number of bits exchanged.

This model and its bounded-error probabilistic version have proved a useful tool in obtaining area–time tradeoffs for VLSI computation ([Th], [Ya2]). Nondeterministic protocols were introduced by Lipton and Sedgewick [LS], mainly because it was apparent that the known lower bound techniques for deterministic protocols worked for nondeterministic ones as well. (Although the

0272-5428/86/0000/0337$01.00

matrix rank lower bound used by Mehlhorn and Schmidt [MS] applies only to deterministic protocols, most lower bound proofs do work for nondeterministic protocols.) This phenomenon was explained by the surprising result of Aho, Ullman and Yannakakis [AUY] that if *both f and its negation* have nondeterministic protocols of length t then f has a deterministic protocol of length $O(t^2)$. As pointed out by Papadimitriou and Sipser [PSr], this result is analogous to a $P = NP \cap coNP$ statement, and is the starting point of the present work. The statement corresponding to $P \neq NP$ is comparatively straightforward: checking the relation $x \neq y$ requires n bits information transfer deterministically [Ya1] and only $\log n$ nondeterministically. (In [PSr], the same exponential speedup is proved in the more difficult model where the $2n$ input bits are divided between North and South "optimally". In this model equality becomes trivial to test deterministically; however, an exponential gap remains for the function "triangle-free graph". We shall not consider problems of this model here.) Bounded-error probabilistic protocols were introduced by Yao [Ya3]. They can be exponentially more powerful than deterministic or even nondeterministic protocols: "equality" can be tested with high probability at a cost of only $O(\log n)$ bits of communication ([Ya1], [Ra], [JKS], cf. also [MS]). The other side of the comparison problem between the powers of nondeterministic vs. bounded-error probabilistic protocols will be resolved in this paper: we prove that nondeterminism can be exponentially more powerful than bounded-error probabilistic protocols. Such lower bounds are considerably more difficult to prove than deterministic lower bounds and have obvious potential significance for VLSI. Indeed, nontrivial lower bounds for a wide variety of graph properties follow immediately.

Unbounded-error probabilistic protocols were introduced by Paturi and Simon [PSn]. Lower bound proofs in this model apparently require deeper mathematical tools such as those employed by Alon, Frankl and Rödl [AFR] in their proof that almost every Boolean function requires at least $n - 5$ bits communication. It remains an open problem to exhibit even a single explicit example of a Boolean function requiring more than $\log n$ bits. (By $\log n$ we mean base 2 logarithm throughout.) It is conjectured that "inner product mod 2" ($\sum_{i=1}^{n} x_i y_i \bmod 2$) requires $n - 1$ bits.

2. Complexity classes

Judging from what is commonly deemed "easy" and "difficult", the natural unit by which to measure communication complexity is the quantity $p = \log n$. "Time" being the number of bits exchanged, we shall thus speak of a "polynomial time protocol" if at most $p^c = (\log n)^c$ bits are transferred for some constant c. We can now define the analogues of P, NP, BPP and PP as those classes of languages (having words of even lengths only) admitting polynomial time deterministic, nondeterministic, bounded-error and unbounded-error probabilistic protocols, resp. Note that in our definition, these classes are "non-uniform": to recognize a language L, each level $L_n = L \cap \{0,1\}^{2n}$ has to be recognized by a separate protocol and this sequence of protocols may have no common pattern. ($\{0,1\}^{2n}$ is identified with $\{0,1\}^n \times \{0,1\}^n$ and L_n is identified with the Boolean function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ where $f(x,y) = 1$ iff $(x,y) \in L_n$.)

As we have done in the Abstract, we might denote these classes by P^{cc}, NP^{cc}, etc. For convenience (and added thrill) we shall omit the ^{cc} superscript and (ab)use the familiar notation to obtain such impressive statements as $NP \neq coNP$. Throughout this paper, such notation will refer to communication complexity classes unless TM's are specifically mentioned. Thus $P = NP \cap coNP$ [AUY]. Again the analysis of "equality" [Ya1], [Ra] shows that $P \neq BPP$ (and, consequently, $BPP \not\subseteq NP$). One of our main results is $NP \not\subseteq BPP$. Care has to be exercised in designating the analog of PP. We denote the class defined by polynomial time Paturi-Simon [PSn] unbounded error probabilistic protocols by UPP and reserve the symbol PP for a different class. We shall define $PSPACE$ and find that PP is a subclass of $PSPACE$ while UPP is likely to be incomparable with $PSPACE$.

3. Alternation

For deterministic protocols, Duriš, Galil and Schnitger [DGS] proved that the number of rounds in communication defines a strict hierarchy with exponential increase in power at each step.

We propose the notion of another hierarchy, one that corresponds to alternating Turing machines. The resulting complexity classes will be the members of the *polynomial time hierarchy* in communication complexity, denoted Σ_k and Π_k, $k = 0, 1, \ldots$. "Unlimited" alternation will define $PSPACE$ (although we do not have a notion corresponding to space). Separation of these classes is a major open problem.

In this section we give a definition, in terms of protocols, of the classes in the hierarchy. The equivalent characterization in terms of rectangle-based formulas to be given in the next section is so natural, however (at least once one digested the notion that rectangles are the quintessence of communication complexity theory) that

the reader may consider skipping this section and view the results of the next one as definitions.

As before, in an *alternating communication protocol,* North and South cooperatively try to evaluate $f(x,y)$ where x is known to North, y to South, the Boolean function f is fully known to both and both have unlimited computational power. The difference is that now, North and South are just pawns (or, rather, referees) in an antagonistic game of two much more powerful players, East and West. (Once again, the fate of North-South dialogue depends on the outcome of East-West conflict. This analogy doesn't run very deep, though.) East and West are nondeterministic. In a previously agreed number of alternate moves they write (0,1)-strings of previously agreed lengths on a board viewed by both North and South. After the East–West game, North and South exchange messages according to a deterministic protocol until one of them declares the winner (East or West). This way the Boolean function f is computed if $f(x,y) = 1$ precisely when *East has a winning strategy.* (East is the *existential,* West is the *universal* player.)

The resources used by such a protocol are the total *number of moves* by East and West, the total *length of the moves* (guess strings), and the length of the *evaluating protocol.*

Let us restrict all these resources to be polynomially bounded, i. e. have total length $< (\log n)^c$. If a language L can be recognized by such a *polynomially bounded alternating protocol* then L belongs to *PSPACE.* If, in addition, the total number of moves in the game is $k \geq 1$ and East (West) moves first then L belongs to Σ_k (Π_k, resp.). For reasons to become clear later, we define Π_0 to consist of the *rectangles* (languages L such that for every n, $L_n = L \cap \{0,1\}^{2n}$ is a rectangle, i. e. a set of the form $X \times Y$ where $X, Y \subseteq \{0,1\}^n$) and Σ_0 to consist of the *complements of rectangles.*

The Σ_k and the Π_k form the polynomial time hierarchy. It should be clear that $NP = \Sigma_1$ and $coNP = \Pi_1$, moreover

$$\Sigma_k \cup \Pi_k \subseteq \Sigma_{k+1} \cap \Pi_{k+1} \subseteq PSPACE.$$

Problem 3.1. *Prove that* $\Sigma_2 \neq \Pi_2$.

4. Characterization of the polynomial time hierarchy

We give two more equivalent definitions of the hierarchy. The equivalence follows from the observation that the referees' protocol can be greatly simplified. One can modify the game such that in the end only a *single bit* of (North to South) communication is needed. This information will suffice for South to declare the winner. The reason is that whoever makes the last move in the East–West game, should guess and post the entire subsequent North–South communication. North can then verify the correctness of his part of the communication and acknowledge it by sending a "1" to South. If South, too, finds the communication correct, then he is able to declare the winner. If the guess was incorrect, the player who made the guess loses.

All this goes very near to proving the following characterization of the polynomial time hierarchy.

Let $\Sigma_\infty = \Pi_\infty = PSPACE$. Let $k \geq 1$ including the possibility $k = \infty$.

Let us fix a positive constant c and an integer $k \geq 0$. Let L be a language containing strings of even lengths only, and let $L_n = L \cap \{0,1\}^{2n}$. Let the nonnegative integers $l_1(n), \ldots, l_k(n)$ satisfy the inequality $l(n) = \sum_{i=1}^{k} l_i(n) < (\log n)^c$. If $k = \infty$, replace every occurrence of k as a subscript or limit by $k(n)$ where $k(n) < (\log n)^c$.

Definition 4.1. $L \in \Sigma_k^c$ if, for some choice of the $l_i(n)$, there exist Boolean functions $\varphi, \psi : \{0,1\}^{n+l(n)} \to \{0,1\}$ such that $(x,y) \in L_n$ iff

$$\exists u_1 \forall u_2 \exists u_3 \ldots Q_k u_k (\varphi(x,u) \diamondsuit \psi(y,u))$$

where $u = u_1 \ldots u_k$, $u_i \in \{0,1\}^{l_i(n)}$ and $\diamondsuit$ stands for $\vee$ ("or") if k is even and for $\wedge$ ("and") if k is odd. We define Π_k^c analogously, by switching the roles of the two quantifiers and of the two Boolean operators.

Definition 4.2. $\Sigma_k = \bigcup_{c>0} \Sigma_k^c$, $\quad \Pi_k = \bigcup_{c>0} \Pi_k^c$.

We note that the above definition indeed yields the rectangles (formulas of the form $\varphi(x) \wedge \psi(y)$) for Π_0 and the complements of rectangles ($\varphi(x) \vee \psi(y)$) for Σ_0.

For $L \in \Sigma_k^c$ (Π_k^c), let us call L_n a Σ_k^c-shape (Π_k^c-shape, resp.). Another, recursive definition of Σ_k^c and Π_k^c can be given as follows. A *shape* in dimension n is a subset of $\{0,1\}^{2n}$. Occasionally we think of a shape as a (0,1)-matrix of size $2^n \times 2^n$.

Definition 4.3. Π_0^c-shapes are the rectangles. Σ_k^c-shapes are the complements (in $\{0,1\}^{2n}$) of Π_k^c-shapes. Π_k^c-shapes are the intersections of at most $2^{(\log n)^c}$ Σ_{k-1}^c-shapes. A language L is in Σ_k (Π_k) if for some $c > 0$ each L_n is a Σ_k^c-shape (Π_k^c-shape, resp.).

To obtain *PSPACE* ($k = \infty$), we have to set $k = k(n) = \lfloor (\log n)^c \rfloor$ in the second half of the last line of the definition.

In particular, we recognize the familiar characterization of *NP* underlying most known lower bound proofs: the Σ_1-shapes are unions of $2^{(\log n)^c}$ rectangles.

The following is now immediate.

Proposition 4.4. *These definitions are equivalent to the one given in the previous section.* ♠

5. Rectangular reduction. Completeness.

As before, any language will be assumed to have strings of even lengths only. The first half of any string will be denoted x, the second half y.

Definition 5.1. A *rectangular reduction* from a language L to another, L', is a pair of functions (f, g) mapping (0,1)-strings to (0,1)-strings such that for some constant c, for every n and every pair of strings $x, y \in \{0,1\}^n$,

$$(1) \qquad |f(x)| = |g(y)| < 2^{(\log n)^c};$$

$$(2) \qquad (x,y) \in L \quad \text{iff} \quad (f(x), g(y)) \in L'.$$

We denote this circumstance by $L \sqsubseteq L'$. If this is the case and $L' \in \Pi_k$ then $L \in \Pi_k$; similarly for Σ_k. (This is clear from the definitions of Σ_k and Π_k.)

Definition 5.2. L is *complete* in the class Φ if $L \in \Phi$ and for every $L' \in \Phi$, $\quad L' \sqsubseteq L$.

Next we determine natural complete languages in each member of the polynomial time hierarchy.

We define the language $L(k)$ for every $k \geq 1$ as follows. Set $m = n^{1/k}$. If m is not an integer, make $L(k)_n$ empty. Otherwise let us think of $x, y \in \{0,1\}^n$ as k-dimensional arrays with entries denoted $x[i_1, \ldots, i_k]$ and $y[i_1, \ldots, i_k]$, where $1 \leq i_j \leq m$. Now set

$$(x,y) \in L(k)_n \text{ iff } \exists i_1 \forall i_2 \ldots Q_k i_k (x[i_1 \ldots i_k] \diamond y[i_1 \ldots i_k])$$

where Q_k and $\diamond$ have the same meanings as in Definition 4.1.

For $k = \infty$ we give two different variants. If $n = 2^p$, p an integer, set $m = 2$, $k = p$ in the above definition to obtain the language $L(\infty)$. (Make $L(\infty)_n$ empty if n is not a power of 2.) To define the language $L(\sqrt{\infty})$, assume in addition that $p = q^2$ is a perfect square. Set $m = 2^q$, $k = q$.

That we made the right choice of definitions is confirmed by the following theorem.

Theorem 5.3. *For $k \geq 1$, the language $L(k)$ is Σ_k-complete. Both $L(\infty)$ and $L(\sqrt{\infty})$ are PSPACE-complete.* ♠

Proof. For simplicity, let $k = 2$. Let $L \in \Pi_2$; we want to reduce L to $L(2)$. By definition,

$$(x,y) \in L \quad \text{iff} \quad \exists i \forall j (\varphi(x,i,j) \vee \psi(y,i,j)).$$

Here the range of i and j is bounded by some integer $M < 2^{(\log n)^c}$. By padding, we may assume that both i and j always range through the entire set $\{1, \ldots, M\}$. Let now $n' = M^2$,

$$f(x) := \text{concatenation}_{i,j}(\varphi(x,i,j)),$$

$$g(y) := \text{concatenation}_{i,j}(\psi(y,i,j)).$$

Now, by definition, $(x,y) \in L$ iff $(f(x), g(y)) \in L(2)$.

The proof has to be slightly modified for the case $k = \infty$. ♠

By switching the roles of the two quantifiers and the two Boolean operators we obtain the corresponding Π_k-complete languages $L(-k)$.

These languages remain complete in their corresponding classes if in the above definition we replace either or both of $x[i_1 \ldots i_k]$, $y[i_1 \ldots i_k]$ by their negation.

In particular, the following variant of $L(-1)$ is also *coNP*-complete ($coNP = \Pi_1$):

$$(x,y) \in L'(-1) \quad \text{iff} \quad \forall i (x[i] = 0 \vee y[i] = 0).$$

Regarding x and y as the characteristic vectors of subsets of a set of n elements, this is precisely the *disjointness* problem. The fact that it is *coNP*-complete indicates that investigating the complexity of the disjointness function is of particular interest. The main results of the remaining sections will be devoted to different aspects of this problem.

From the hierarchical point of view, the significance of this family of complete problems is clear. Just as for Turing machine classes, if a Π_k-complete language L belongs to Σ_k then $\Pi_k = \Sigma_k = \Pi_{k+1} = \Sigma_{k+1} = \ldots$.

6. Oracles. The power of "equality"

What is the *relative complexity* of "equality" and "disjointness"? Both problems require $\Omega(n)$ bits of communication with any deterministic protocol and even their nondeterministic behavior seems identical: each requires $\Omega(n)$ bits nondeterministically ("equality requires n bits, "disjointness" $n - O(\log n)$ bits), the negation of each requires only $O(\log n)$ bits nondeterministically. Nevertheless, there is a marked difference between the complexities of the two problems.

Reductions play a crucial role in measuring the relative complexity of different languages. While "disjointness" is *coNP*-complete and therefore "equality", a member of *coNP*, has a rectangular reduction to it (in fact an easy one), the converse is not true:

Observation 6.1. *"Equality" is not coNP-complete.*

This fact, however, is too obvious to be convincing. One can easily show that even the following trivial language L has no rectangular reduction to "equality". Let $(x,y) \in L$ if $x_1 = 1 \vee y_1 = 1$. (This is the complement of a rectangle. Note that the deterministic one-way communication complexity of L is a single bit.) What this indicates is that the notion of rectangular reduction is too restrictive. (Yet, there *exist* complete problems in all our classes!)

A more flexible notion of reducibility is required and is provided by the following analogue of Turing (Cook) reducibility.

As before, a *language* will always consist of pairs (x, y) of strings of equal length. Let L be the language to be used as an *oracle* and (x, y) the input. Our objective is to determine whether or not $(x, y) \in L'$ for a given language L'.

Definition 6.2. An *oracle-query* is a question of the form "$(f(x), g(y)) \in L$?", where f and g are $\{0,1\}^* \to \{0,1\}^*$ functions such that $|x| = |y|$ implies $|f(x)| = |g(y)|$. The query is specified by the pair (f, g). The *length* of the query is the common length of the strings $f(x)$ and $g(y)$.

Definition 6.3. A *pure oracle-reduction* of L' to L of length m is a strategy of asking a sequence of m oracle queries, each query depending on the string of previous responses but independent of the input (x, y), such that membership of (x, y) in L' is a Boolean function of the string of responses. The *complexity* of a sequence of queries (f_i, g_i) is $\sum_{i=1}^{m} \log |f_i(x)|$. We say that the complexity of the reduction is the function $F(n)$ if $F(n)$ is the maximum complexity of the query sequence over all pairs of strings of length $\leq n$.

Thus in a *polynomial time reduction*, the number of queries is $\leq (\log n)^c$ and each query has length $\leq 2^{(\log n)^c}$.

North and South play no role in this notion of reduction. They do in the following, more powerful one.

Definition 6.4. An *oracle protocol* is a deterministic communication protocol between North and South, allowing each party to query the oracle according to a predetermined strategy. The queries and their timing may depend on the information available to each party, including their part of the input. The complexity of such a protocol is the complexity of the query sequence as defined above plus the number of bits exchanged.

The distinction between pure oracle reductions and oracle-protocol reductions does not seem to have a Turing-machine analog.

We shall also need a weaker reduction concept, borrowed from recursion theory.

Definition 6.5. A *truth-table reduction* is a pure oracle-reduction where the queries do not depend on the responses to earlier queries (the sequence of queries is fixed in advance).

Note that a rectangular reduction is an oracle (truth table) reduction with a single query directly answering the membership question in L'. Let us also observe that for any L' and any non-trivial oracle, $2n$ queries of constant length suffice to find out what the input strings are, hence $O(n)$ is an upper bound on the complexity needed for truth-table reductions. (A language L is *trivial* if for every n, membership in L_n depends either on x or on y only.)

Conjecture 6.6. *Any oracle-protocol reduction of "disjointness" to "equality" requires $\Omega(n)$ oracle queries.*

We have two results in this direction.

Theorem 6.7. *Any truth-table reduction of "disjointness" to "equality" requires $\Omega(\sqrt{n})$ queries.*

Proof. The proof of this result is by a reduction/elimination process. We have to generalize the result and prove, by induction on K, that with K queries, not only can the truth-table not give the correct answer for *every* (x, y) but even for every x coupled with every member y of any set $Y \subseteq \{0,1\}^n$ where $|Y| > n^K 2^{cK^2}$. We write the Boolean function evaluating the responses to the queries in disjunctive normal form. At each step we assign truth values to some pair x_i, y_i and remove a portion of $|Y|$, thereby eliminating either certain kinds of clauses (e.g. all clauses containing only negated equalities) or certain queries, still leaving the same kind of problem of not too much smaller size.

We describe the process. We assume that for any x and for any $y \in Y$, x and y are disjoint precisely if a disjunction of certain clauses is true. Each clause is a disjunction of primary relations. Three kinds of primary relations are permitted: Boolean functions depending on x only, equalities of the form $f_j(x) = g_j(y)$, and the negated equalities. We divide the clauses into three categories. Type "0" clauses depend on x only. (We may assume there is at most one such clause.) Type "1" clauses contain no equality, at least one negated equality and possibly a function of x. The rest are type "2": each must involve at least one equality. Let K be the number of pairs of functions (f_j, g_j) involved. (Both the equality and the negated equality of each pair may be involved in several clauses.)

Reduction 0. If there is a clause of type "0", pick an i, $1 \leq i \leq n$ such that $y_i = 1$ for some $y \in Y$ and $y_i = 0$ for at least $|Y|/n$ members of $|Y|$. Reduce n and Y by restricting the problem to the set $x_i = 1, y_i = 0$. (* This will eliminate the type "0" clause if there was one.*)

Reduction 1a. Set $Y_i^\epsilon = \{ y \in Y : y_i = \epsilon \}$ where $\epsilon \in \{0,1\}$. Remove from Y the set $\bigcup \{ Y_i^1 : |Y_i^1| < |Y|/2n \}$ if this operation turns *all* type "1" clauses into type "0". A type "1" clause $A(x, y) = a(x) \wedge \bigwedge_j (f_j(x) \neq g_j(y))$ is reduced to $a(x)$ (type "0") if for every x and every $y \in Y$, $a(x)$ implies that x and y are disjoint.

Reduction 1b. If there is a type "1" clause, select $x = u$, subscript i and a type "1" clause $A(x, y) = a(x) \wedge \bigwedge_{j=1}^{m} (f_j(x) \neq g_j(y))$ such that $|Y_i^1| \geq |Y|/2n$, $u_i = 1$,

and $a(u) = 1$. Set $Z_j = \{ y \in Y_i^1 : f_j(u) = g_j(y) \}$. Select j such that $|Z_j| \geq |Y_i^1|/m$. Reduce Y to Z_j (in particular, set $y_i = 1$), reduce n to $n-1$ by setting $x_i = 0$ and replace all occurrences of $g_j(y)$ by the constant $f_j(u)$, reducing K by at least 1.

Reduction 2. (* Now, all the clauses are of type "2"*) Let $B_j(x,y)$ denote a clause and set $W_j = \{ y \in Y : B_j(0,y) = 1 \}$. Reduce Y to the largest of the W_j. For each equality relation $f(x) = g(y)$ occurring in B_j (we know there is at least one), replace all occurrences of $g(y)$ by $f(0)$, reducing K by at least 1.

In order to justify the process, we first we observe that

$$|Y - \bigcup \{Y_i^0 | Y_i^1 \neq \emptyset\}| \leq 1.$$

From this it follows that a subscript i appropriate for Reduction 0 always exists. Since $y_i = 1$ for some $y \in Y$, the type "0" clause must be identically 0 for every x with $x_i = 1$.

Suppose now that there exists a type "1" clause and Reduction 1b cannot be carried out. Let Y' denote the result of Reduction 1a. Then, for each type "1" clause $A(x,y) = a(x) \wedge \ldots$, and for each x and each $y \in Y'$, if $a(x) = 1$ and $x_i = 1$ then $y_i = 0$, i.e. if $a(x) = 1$ then x and y are disjoint, making the rest of $A(x,y)$ redundant. Therefore Reduction 1a will be carried out.

It is now immediate that at least one out of any four consecutive steps of this procedure will be either Reduction 1b or Reduction 2, thus reducing the value of K and making induction possible.

The cost of each reduction step is a reduction of the size of Y. This is by a factor of at most n in Reduction 0, by 2 in Reduction 1a, by $2nm \leq 2nK \leq n^2$ in Reduction 1b (if $K > n/2$, there is nothing to prove), and by a factor of 2^K at worst in Reduction 2. ♠

A slightly weaker, $\Omega(\sqrt{n}/\log n)$ lower bound for the strongest (oracle-protocol) reduction will follow from the main result of the next section. The techniques of the two proofs are entirely different.

A number of other natural reducibility questions arise, the most intriguing being the strong separation of the levels of the hierarchy: no polynomial time oracle-protocol can reduce $L(k+1)$ to $L(k)$.

7. *BPP* and the polynomial time hierarchy

Bounded-error (two-way) probabilistic protocols (B2PP's) have been defined by Yao [Ya3]. They differ from deterministic protocols in allowing the messages depend on coin-flips. The number of coin-flips is added to the complexity. An input is accepted if the probability of acceptance is at least $1 - \epsilon$ for some fixed ϵ, $0 \leq \epsilon < 1/2$, rejected if the probability of acceptance is at most ϵ, and all input pairs (x,y) must fall in one of these categories. The complexity on an input (x,y) is the average, over all coin flip sequences, of the length of the protocol. The complexity of the language is the maximum of this over all inputs. Let $L \in BPP$ if L is accepted by a polynomial $((\log n)^c)$ time B2PP.

B2PP's can be exponentially more powerful than even nondeterministic protocols: "equality" can be tested in $O(\log n)$ [Ya3], [Ra], [JPS]. For completeness, let us describe this protocol.

NORTH: Picks random prime p, $2 \leq p < 2n$. Transmits p and $x \bmod p$.

SOUTH: Outputs "not equal" if $x \not\equiv y \bmod p$, "equal" otherwise.

Clearly, the "not equal" answer is always correct. The "equal" answer will fail with probability $< 1/3$, because the product of all primes $< n$ is $e^{n(1+o(1))}$.

This proves that $BPP \neq P$ and $BPP \not\subseteq NP$. Our main separation result states that BPP and NP are in fact incomparable.

Theorem 7.1. *$NP \not\subseteq BPP$.*

In order to prove this, we give an exponential (in $\log n$) lower bound for the complexity of the *coNP*-complete problem "disjointness".

Theorem 7.2. *The bounded-error probabilistic complexity of "disjointness" is $\Omega(\sqrt{n})$.*

No nonlogarithmic lower bound for "disjointness" appears to have been known. We derive Theorem 7.2 in Section 8.

Conjecture 7.3. *The bounded error probabilistic complexity of "disjointness" is $\Omega(n)$.*

Proving Conjecture 7.3 would be quite significant since "disjointness" has a linear time rectangular reduction to essentially *any nontrivial monotone graph property* under suitable definitions. We note that, in particular, Theorem 7.2 has this

Corollary 7.4. *The bounded error probabilistic communication complexity of the following problems for sparse n-vertex graphs is $\Omega(\sqrt{n})$: connectedness, planarity, bipartiteness, existence of perfect matching.*

(Sparse means it has $O(n)$ edges.) No probabilistic lower bounds for these problems appear to have been known previously. For connectedness and matching, this is the best lower bound we know, for the others see 9,6.

Let us call a graph property *nontrivial* if for arbitrarily large values of n, there exists a graph with n edges that has the property but none of its proper subgraphs on the same vertex set does (e.g. connectedness, nonplanarity,

non-bipartiteness, existence of perfect matching). We say that a graph property is *invariant under doubling edges* if adding an edge parallel to an existing edge does not change the truth value. Each of the properties mentioned satisfies this condition. Corollary 7.4 is thus, in essence, a particular case of

Corollary 7.5. *The bounded error probabilistic communication complexity of any nontrivial monotone graph property which is invariant under doubling edges is* $\Omega(\sqrt{n})$ *where* n *is the number of edges.*

Proof. Let X be a graph with n edges and minimal with respect to the given property. Let us double every edge of X and assign a pair of Boolean variables x_i, y_i to each pair of parallel edges. Corresponding to any truth assignment to the x_i and y_i there will be a graph $X(x,y)$. Clearly, this graph will have the given property precisely if for every i, at least one of the edges is present, i.e. if the negations of x and y are disjoint. ♠

Another corollary to Theorem 7.2 represents a step toward Conjecture 6.6.

Corollary 7.6. *Any oracle-protocol reduction of "disjointness" to "equality" has complexity* $\Omega(\sqrt{n})$.

Proof. The equality oracle can be replaced by the B2PP of Rabin and Yao. The cost of query (f,g) using this protocol will be $O(\log|f(x)|)$, proportional to the cost charged for an oracle query by Definition 6.3. Therefore the deterministic oracle-protocol complexity of "disjointness" with respect to an "equality" oracle is not less than the B2PP complexity of "disjointness". ♠

Let us remark that in using an equality oracle, the queries (f,g) may be assumed to have length $|f(x)| = |g(y)| \leq n+1$ because $|\{f(x), g(y) : x,y \in \{0,1\}^n\}| \leq 2^{n+1}$. This observation together with Corollary 7.6 implies

Corollary 7.7. *Any pure oracle-reduction of "disjointness" to "equality" requires* $\Omega(\sqrt{n}/\log n)$ *oracle queries.* ♠

An adaptation to communication complexity of the Sipser-Gács-Lautemann proof [Si1], [La] yields

Proposition 7.8. $BPP \subset \Sigma_2 \cap \Pi_2$. ♠

This observation further confirms our choice of definitions.

8. Strong distributional complexity of "disjointness"

The *ϵ-error distributional complexity* $D_\epsilon(f)$ of a Boolean function $f(x,y)$ is the minimum length of a deterministic protocol correctly computing $f(x,y)$ on all but an ϵ fraction of inputs. Yao [Ya1] observes that $2C_\epsilon(f) \geq D_{2\epsilon}(f)$ where $C_\epsilon(f)$ denotes the ϵ-error B2PP complexity of f. This inequality continues to hold if we restrict the domain of f on the right hand side, or, more generally, introduce an arbitrary probability measure μ on $\{0,1\}^{2n}$. If we draw the input pairs (x,y) at random according to the measure μ, we obtain the distributional complexity $D_\epsilon(f|\mu)$.

Definition 8.1. A probability measure μ on $\{0,1\}^{2n}$ is *rectangular* if it is a product $\lambda \times \rho$ of probability measures on $\{0,1\}^n$. The measure of a rectangle is thus $\mu(X \times Y) = \lambda(X) \times \rho(Y)$. – This means we pick x and y *independently* from two arbitrary probability distributions.

Definition 8.2. The *strong ϵ-error distributional complexity* of f, $SD_\epsilon(f)$ is the supremum of $D_\epsilon(f|\lambda \times \rho)$ over all rectangular measures $\mu = \lambda \times \rho$ on $\{0,1\}^{2n}$.

Clearly, $2C_\epsilon(f) \geq SD_{2\epsilon}(f)$.

We have nearly tight bounds for the "disjointness" function d.

Theorem 8.3. *For any* $\epsilon < 1/100$, $\Omega(\sqrt{n}) \leq SD_\epsilon(d) \leq O(\sqrt{n}\log n)$.

The lower bound implies Theorem 7.2. The upper bound indicates that distributional complexity will be of little use for improving the lower bound of Theorem 7.2.

Next, we outline the *proof of the upper bound.* We view (0,1)-strings as subsets of the set $[n] = \{1,\ldots,n\}$. The protocol will refer to a huge database. For every subset v of the universe $[n]$, and for every k, $1 \leq k \leq n$, let $W(v,k)$ denote the family of sets $\{w \subseteq [n] : |w \cap v| = k\}$. For each $|v|, k, l \geq \sqrt{n}$, "$\lambda-$ and $\rho-$representative" subsets $L(v,k,l) \subseteq W(v,k)$ and $R(v,k,l) \subseteq W(v,l)$ are selected in advance. Each of these sets must have $r = O(1/\epsilon^2)$ members and have the property that if the conditional probability

$$p(v,k,l) = \mu_{x,y}\{x \cap y = \emptyset | x \in W(v,k), y \in W(v,l)\}$$

is at least p where $p = \Omega(\epsilon)$ is a constant, then

$$\rho_y\{y | y \in W(v,l) \wedge (\forall x \in L(v,k,l) \quad x \cap y = \emptyset)\} \leq O(\epsilon)$$

The analogous condition must hold for the $R(v,k,l)$. (The existence of such families can be shown by a probabilistic argument.)

We sketch the protocol. The protocol will have phases. Each phase corresponds to a subset $v \subseteq [n]$, known to both parties, where $x - v$ and $y - v$ are disjoint. The objective of each phase is either to determine whether or not $x \cap v$ and $y \cap v$ are disjoint or to reduce the size of v by at least $\sqrt{n}$. Initially $v = [n]$. First thing in each phase, North and South inform each other of the cardinalities of their respective sets: $k = |x \cap v|$ and $l = |y \cap v|$. Suppose $k \leq l$. If $k \leq \sqrt{n}$ then North transmits $x \cap v$ (this

requires $O(\sqrt{n}\log n)$ bits) from which South determines the output. Otherwise, if $p(v,k,l) < p$ then we output "not disjoint". (We may err here.) Otherwise South selects a member x_i of $L(v,k,l)$ which is disjoint from $v \cap y$ and transmits i to North (constant number of bits). If no such i is found, report "not disjoint" and terminate. (We may err again.) Change v to $v - x_i$, start the next phase. ♠

Next we sketch the *lower bound proof.* Yao's technique [Ya3] requires the conditions of "moderateness" (for a random pair (x,y), the probability of $f(x,y)$ should be bounded away from 0 and 1) and "anticorrelatedness" (for any given x,y and a random $z \subset [n]$, the events $f(x,z)$ and $f(y,z)$ should not reinforce each other by more than a factor of $(1+2^{-cn})$). For "disjointness", these two conditions cannot simultaneously be satisfied for any rectangular probability measure. We shall select our measure to satisfy "moderateness" and make up for the absence of "anticorrelatedness" by a combinatorial argument.

Let $X = Y$ consist of all subsets of size $\sqrt{n}$ of $[n]$ (without loss of generality, assume n is a perfect square, divisible by 12). We shall select the pairs (x,y) at random from the uniform distribution over $X \times Y$. A random pair (x,y) now has probability $\approx 1/e$ to be disjoint. An *ϵ-error 1-rectangle* is a set $R = F \times G$ where $F \subseteq X$, $G \subseteq Y$ such that $f(x,y) = 1$ (i.e. $x \cap y = \emptyset$) on all but an ϵ fraction of R. Following [Ya3], we only have to prove that $|R| < |X||Y|2^{-c\sqrt{n}}$.

Let F_1 consist of those $x \in F$ satisfying $|\{y : x \cap y \neq \emptyset\}| < 2\epsilon|G|$. Clearly $|F_1| \geq |F|/2$.

Proposition 8.4. *Given any $x_1,\ldots,x_k \in F_1$, at most $|G|/2$ of the $y \in G$ intersect more than $4\epsilon k$ of the x_i.* ♠

Lemma 8.5. *If $|F| \geq |X|2^{-c\sqrt{n}}$ then there exist $x_1,\ldots,x_k \in F$ such that $k \geq \sqrt{n}/3$ and for every $l \leq k$,*

$$|x_l \cap \bigcup_{i<l} x_i| < \sqrt{n}/2.$$

Proof. Select the x_i inductively. Suppose $x_1 \ldots x_{l-1}$ have been selected and that $z = \bigcup_{i<l} x_i$. We infer $|z| < l\sqrt{n} < n/3$. The number of those $x \in X$ satisfying $|z \cap x| > \sqrt{n}/2$ is therefore less than

$$n\binom{n/3}{\sqrt{n}/2}\binom{2n/3}{\sqrt{n}/2} < \binom{n}{\sqrt{n}}2^{-c\sqrt{n}}.$$

Therefore $|x_l \cup z| < \sqrt{n}/2$ for some $x_l \in F_1$. ♠

Now, combining 8.4 and 8.5 we obtain an upper bound for $|R|$ as follows. If the condition in 8.5 does not hold, we are done. Otherwise, there are at most $\binom{k}{4\epsilon k}$ ways to select those $4\epsilon k$ of the x_i which a given $y \in G$ is allowed to intersect. The union of the remaining x_i has size $> k(1-4\epsilon)\sqrt{n}/2 > k\sqrt{n}/3 \geq n/9$. Therefore

$$|G| < 2\binom{k}{4\epsilon k}\binom{8n/9}{\sqrt{n}} < 2^{-c\sqrt{n}}\binom{n}{\sqrt{n}} = 2^{-c\sqrt{n}}|Y|,$$

and again we conclude that $|R| < |X||Y|2^{-c\sqrt{n}}$. ♠

9. *PSPACE*, $\#P$ and the *PP*-clones

We can generalize our model by considering the computation of functions with ranges other than $\{0,1\}$: as before, we require that South post the result. We count the length of the output as part of the protocol. We shall denote by FP, $F\Sigma_i$, and $FPSPACE$ the classes of functions computable by polynomial time protocols that are deterministic, Σ_i, and $PSPACE$.respectively. Note that the length of the output in these cases must be bounded by $(\log n)^c$ for some c. Non-boolean functions have been studied before in communication complexity [EP],[Ab].

Consider the function $H(x,y)$ =the Hamming distance between x and y.

Proposition 9.1. *Any deterministic protocol for $H(x,y)$ requires $n + \log n$ bits (that are also sufficient).*

Proof. See [EP]. ♠

Proposition 9.2 . *$H(x,y) \in PSPACE$.*

Proof. We sketch an $FPSPACE$ protocol for H: for simplicity, assume that the common length of x and y is 2^p for some natural number p. In round i, $0 \leq i \leq p-1$, the existential player will guess the Hamming distance d_i between two substrings of the input: a certain initial substring of x, of length $2^p - i$, and the corresponding substring in y. It will also guess the Hamming distances $d_{i,0}$ between the left half of x and the corresponding substring of y, and $d_{i,1}$, the Hamming distance between the right halves, subject to $d_i = d_{i,0} + d_{i,1}$. It sends the pair $d_{i,0}, d_{i,1}$) to the existential player, who challenges either the left or the right half by sending back a 0 or a 1. The protocol continues on the selected substring. In the last round, the existential player simply sends the bit selected. Clearly all guesses can be verified only if they are correct, in which case the Hamming distance is d_0. The total length of the protocol is $O(p^2)$. ♠

Another interesting class of functions is $\#P$ that we now define. Let f be a language in NP and consider a corresponding NP-protocol. For each (x,y), let $F(x,y)$ count the number of guesses posted by *East* that lead to acceptance of (x,y). An example of a function in $\#P$ is the "inner product" (IP) function $\sum_{i=1}^{n} x_i y_i$. (Consider the nondeterministic protocol that selects a bit of x and

sends the bit and its address. If both the bit sent and the corresponding bit of y are 1, the pair is accepted. The number of accepting computations is the inner product and the length of the protocol is clearly $O(\log|x|)$.) Note that this language is in fact $\#P$-complete, since it is the counting problem associated with the NP-complete "nondisjointness".

One can define classes of languages associated with some of these function classes. They are again analogs of well-known complexity classes, but there seems to be a great variety of possible definitions. While in some cases we can show classes to be distinct, there are many open questions about the relationships among these families of languages.

The language class associated with $\#P$ is $P^{\#P}$ (deterministic polynomial time protocols with a $\#P$ oracle). There are many natural counting problems in this class (e.g. $\{([x,k],y)$ — the inner product of x and y is exactly $k\}$.

A related class of languages, which we shall call PP, arises by counting accepting guess-strings in an NP-protocol. Let us assume that all guess strings have equal length and let $f(x,y)=1$ iff more than half of the guesses lead to acceptance. Of course, this can be interpreted as assigning probability 2^{-l} to a message of length l, and requiring that the protocol succeed with probability greater than 1/2. Note that here, the probability of error is less than $1/2-2^{-(\log n)^c}$, "moderately bounded away" from 1/2. We don't know, however, if this boundedness condition itself would suffice to put the language in PP, raising the possibility of yet another interesting related class.

The largest of the probabilistic classes, which we call UPP, was defined by *unrestricted-error probabilistic protocols* in [PSn]: the protocol choses the appropriate message based on the input, messages exchanged so far, and a probability distribution. Of course, every message must be at most $(\log n)^c$ long As in the case of PP, the language accepted consists of the pairs for which the protocol succeeds with probability greater than 1/2. Since there are no restrictions on the probabilities used by the protocol, we cannot guarantee that the probability of acceptance will be bounded away from 1/2.

We have considered previously the class BPP, where the accepting and rejecting probabilities are well separated. Note that there is no loss of power if we restrict BPP protocols to be of length $(\log n)^c$.

There are many interesting questions about the relationships among these complexity classes. It is easy to see that $PP \subseteq UPP$. Also, $BPP \subseteq PP \subseteq P^{\#P} \subseteq PSPACE$. The only nonobvious inclusion in the chain above is that $BPP \subseteq PP$ since BPP is defined by *two-way* protocols. However clearly $BPP \subseteq$ *2-way* PP and *2-way* $PP = PP$.

We believe that UPP is not comparable to either $PSPACE$ or $P^{\#P}$.

The properties of the inner product function IP deserve further study. IP is $\#P$-complete (it is the counting problem associated with the NP-complete "nondisjointedness"). Hence Vazirani's $\Omega(n/\log n)$ lower bound [Va] for the B2PP complexity of $IP2$, the inner product modulo 2, implies $BPP \neq P^{\#P}$. The lower bound on $IP2$ has been improved to the sharp $\Omega(n)$ by an elegant argument in [CG] (Theorem 10). Their proof is essentially based on the following appealing lemma due to J.H. Lindsey ([ES] p. 88). We state it here because apparently it has not been explicitly stated elsewhere. Recall that a Hadamard matrix is a square matrix with +1 and -1 entries whose rows are pairwise orthogonal.

Lemma 9.3. *Let H be an $m \times m$ Hadamard matrix and T an arbitrary $a \times b$ submatrix of H. Then the difference between the number of +1's and -1's in T is at most $\sqrt{abm}$.*

Proof. Let $H=(h_{i,j})$. We may assume that T consists of the first a rows and b columns. Let v_i denote the i-th row of H. By orthogonality

$$(\sum_{i=1}^{a} v_i)^2 = \sum_{i=1}^{a} v_i^2 = am.$$

Now take

$$(\sum_{i=1}^{a}\sum_{j=1}^{b} h_{i,j})^2 \leq b\sum_{j=1}^{b}(\sum_{i=1}^{a} h_{i,j})^2$$

(the Cauchy-Schwarz inequality)

$$\leq b\sum_{j=1}^{m}(\sum_{i=1}^{a} h_{i,j})^2 = b(\sum_{i=1}^{a} v_i)^2 = abm.\spadesuit$$

Corollary 9.4. *The B2PP complexity of "inner product mod 2" is $\Omega(n)$.* $\spadesuit$

It would be interesting to clarify the status of this function with respect to our web of complexity classes. The conjecture $IP2 \not\subseteq UPP$ ([PSn], [AFR]) would imply $P^{\#P} \not\subseteq UPP$, and thus $PP \neq P^{\#P}$.

Some other intriguing questions include the relationship of the polynomial hierarchy to these classes. We observed in Proposition 7.8 that $BPP \subset \Sigma_2 \cap \Pi_2$, trivially $NP \subset PP$ (and also $coNP \subset PP$, so the inclusion is proper), but we know of no other inclusions or differences. For example, is $\Sigma_2 \subset UPP$?

Other natural classes include $\# - 2P$, the class of problems for which an NP protocol has an even number of successful computations, U, for which it has a unique one. Clearly $\# - 2P \subset P^{\#P}$, but we do not know the relationship between PP and $\# - 2P$. Note that the language $IP2$, "inner product modulo 2" is complete for $\# - 2P$, so we know that $\# - 2P \not\subset BPP$.

There are several interesting hierarchies other than the polynomial hierarchy. For example, consider protocols where East has nondeterministic and West random choices. Denote by M the "majority" quantifier, and consider the languages L_i' defined by $(x,y) \in L_i'$ iff $\exists u_1 M u_2 \exists ... Q_i u_i (\phi(x,u) \diamond \psi(y,u))$ where u, ϕ, Q_i and ψ are as in definition 4.1. These languages are complete for certain randomized games, and the complete languages L_i can be reduced to them. Are the corresponding classes new? One can easily invent other such hierarchies, by using consecutive M quantifiers, by reproducing the Arthur-Merlin paradigm [B], and so on. It is conforting to note that $PSPACE$ still includes these hierarchies, but we do not know of interesting proper inclusions.

Finally, as an "application" of our theory, observe that reductions from $L(\infty)$ prove the following, somewhat surprising fact:

Theorem 9.5. *The following graph problems are PSPACE-hard: undirected graph reachability, planarity, bipartiteness, 2-CNF-satisfiability.*

The reduction uses a series parallel network between two vertices a and b in the undirected graph. ♠

Observing that the inner product mod 2 $IP2 \in P^{\#P} \subseteq PSPACE$ and using the $IP2$ lower bound we were led to the following

Corollary 9.6. *The bounded-error probabilistic complexity of each of the mentioned problems for sparse graphs on n vertices is $\Omega(n)$.* ♠

This, of course, can then be derived by simple direct reductions from the $IP2$ bound, but we should point out that the structure of complexity classes introduced in this paper was a helpful guide. Corollary 9.6 has the VLSI consequence of $AT^2 = \Omega(n^2)$.

10. Concluding remarks

We have introduced a variety of complexity classes, enabling a classification of some of the previously considered problems as complete, hard for or included in complexity classes of widely varying logical complexity. Separation of these classes seems a major problem deserving considerable effort. Some dividends in terms of VLSI lower bounds are possible but the major benefit seems analogous to oracle-separation results of Turing machine classes [Ya4]. It provides analogies and hopefully useful techniques as well.

References

[AUY] Aho, A.V., J.D.Ullman, and M. Yannakakis, On Notions of Information Transfer in VLSI Circuits, *Proc. 15th STOC*, 1983, pp. 133-139

[AFR] Alon, N., P. Frankl and V. Rödl, Geometric realization of set systems and probabilistic communication complexity, *Proc. 26th IEEE FOCS*, Portland OR 1985, pp. 277-280

[B] Babai, L., Trading Group Theory for Randomness, *Proc. 17th STOC*, 1985, pp. 421-429

[BGS] Baker, T., J. Gill and R. Solovay, Relativizations of the $P =? NP$ Question, *SIAM J. on Computing* **4** (1975), 431-452

[BS] Baker, T., and A. Selman, A Second Step Toward the Polynomial Hierarchy, *Theoretical Comp. Sci.* **8** (1979), 177-187

[CKS] Chandra, A.K., D. Kozen and L.J. Stockmeyer, Alternation, *JACM* **28** (1981), 114-133

[CG] Chor, B. and O. Goldreich, Unbiased Bits from Sources of Weak Randomness and Probabilistic Communication Complexity, *Proc. 26th IEEE FOCS*, Portland OR 1985, pp. 429-442

[DGS] Duriš, P., Z. Galil and G. Schnitger, Lower Bounds on Communication Complexity, *Proc. 16th STOC*, Washington D.C. 1984, pp. 81-91

[EP] El Gamal, A. and K. F. Pang, Communication Complexity of Computing the Hamming Distance, preprint Stanford U. 1980

[ES] Erdös, P. and J. Spencer, *Probabilistic Methods in Combinatorics*, Academic Press, New York and London, 1974

[FSS] Furst,M., J. Saxe and M. Sipser, Parity, Circuits, and the Polynomial Time Hierarchy, *Proc. 22nd FOCS*, 1981, pp. 260-270

[Gi] J. Gill, Computational complexity of probabilistic Turing machines, *SIAM J. Comp.* **6** (1977), 675-695

[JPS] JaJa, J., V.K. Prasanna Kumar and J. Simon, Information Transfer Under Different Sets of Protocols, *SIAM J. on Computing*, **13** (1984) 840-849

[LLS] Ladner, R., N. Lynch, and A. Selman, A Comparison of Polynomial-Time Reducibilities, *Theoretical Comp. Sci.* **2** (1975), 103-123

[La] Lautemann, C. ,*BPP* and the polynomial hierarchy, *Info. Proc. Letters* **17** (1983), 215-217

[LS] Lipton, R. and R. Sedgewick, Lower bounds for VLSI, *Proc. 13th ACM STOC*, Milwaukee WI 1981, pp.300-307

[MT] Manber, U. and M. Tompa, Probabilistic, Nondeterministic and Alternating Decision Trees, *Proc. 14th ACM STOC*, San Francisco 1982, pp. 234-244

[MS] Mehlhorn, K. and E. M. Schmidt, Las Vegas is better than determinism in VLSI and distributive computing, *Proc. 14th ACM STOC*, San Francisco 1982, pp. 330-337

[Pa] Papadimitriou, C.H., Games against Nature, *Proc. 24th IEEE Symp. Found. Comp. Sci.*, Tucson AZ, 1983, pp. 446-450

[PSr] Papadimitriou, C.H. and M. Sipser, Communication Complexity, *Proc. 14th ACM STOC*, San Francisco 1982, pp. 330-337

[PSn] Paturi, R. and J. Simon, Probabilistic communication complexity, *Proc. 25th IEEE FOCS*, Florida 1984, pp. 118-126

[Ra] Rabin, M., unpublished

[Si1] Sipser, M., A complexity theoretic approach to randomness, *Proc. 15th ACM Symp. on Theory of Comp.*, Boston 1983, 330-335

[Si2] Sipser, M., Borel Sets and Circuit Complexity, *Proc. 15th ACM STOC*, Boston 1983, pp. 61-69

[Sch] Schnitger, G., unpublished manuscript

[St] Stockmeyer, L., The Polynomial-Time Hierarchy, *Theoretical Comp. Sci.* **3** (1977) 1-22

[Th] Thompson, C.D., Area-time complexity for VLSI, *Proc. 11th ACM STOC*, Atlanta GA 1979, pp. 81-88

[Va] Vazirani, U.V., Towards a strong communication complexity theory or generating quasi-random sequences from two communicating slightly-random sources, *Proc. 17th ACM STOC*, Providence RI 1985, pp. 366-378

[Ya1] Yao, A. C.-C., Some Complexity Questions Related to Distributive Computing, *Proc. 11th ACM STOC*, Atlanta GA 1979, pp. 209-213

[Ya2] Yao, A. C.-C., The entropic limitations on VLSI computation, *Proc. 13th ACM STOC*, Milwaukee WI 1981, pp. 308-311

[Ya3] Yao, A. C.-C., Lower Bounds by Probabilistic Arguments, *Proc. 24th IEEE FOCS*, Tucson AZ 1983, pp. 420-428

[Ya4] Yao, A. C.-C., Separating the Polynomial-Time Hierarchy by Oracles, *Proc. 26th IEEE FOCS*, Portland OR 1985, pp.1-10

[Wr] Wrathall, C., Complete Sets and the Polynomial-Time Hierarchy, *Theoretical Comp. Sci.* **3** (1976), 23-33