The authors have chosen to omit some topics that would require harder mathematics (e.g., the connection to Mobius functions [**?**, **?**]). They often give one or two examples, but no more. This approach has kept the book short and readable. There are few wasted words.

I spend most of my Thanksgiving and Christmas vacation (and some time in between) reading this book. That I wanted to speaks well of its choice of topics. That I was able to speaks well of its presentation.

4 Errata

There is a website of errata at http://www.cs.technion.ac.il/~eyalk/book.html

2.2 Chapters 5,6,7: Other Models of Communication

The models of communication used in Chapters 1–4 are simple. Because of the simplicity of the model, the results in those chapters are of limited use in obtaining lower bounds for other (non-communication based) models of computation. In Chapters 5,6, and 7 the book investigates models that are not quite as simple as those in Chapters 1,2,3, but can be used to obtain lower bounds in other fields of theory. We give some highlights of these models and the lower bounds they are used to prove. Since we will do this, we will not summarize Chapters 8–14 on applications.

Let $S \subseteq \{0,1\}^n \times \{0,1\}^n \times Z$. ($Z$ is any set.) Assume Alice has $x \in \{0,1\}^n$ and Bob has $y \in \{0,1\}^n$. Alice and Bob would like to find *some* z such that $(x,y,z) \in S$. If no such z exists then we do not care what Alice and Bob come up with. $D(S)$ is the number of bits Alice and Bob have to exchange to come up with some z. The complexity of relations can be tied very closely to the depth of a circuit. Let f be a monotone function. Let S be the relation

$$\{(x,y,i) \mid f(x) = 0 \wedge f(y) = 1 \wedge x_i = 0 \wedge y_i = 1\}.$$

Then $D(S)$ is exactly the same as the optimal depth of a monotone circuit for f. In Chapters 10 and 11 this relation, along with lower bounds for $D(S)$ for various S, are used to prove lower bounds for monotone circuit depth. The problems considered include s-t-connectivity, Matching, and Set Cover. The work on the Complexity of Relations was done for the purpose of obtaining lower bounds on circuits; hence its application to such is not surprising.

What happens if there are more than two people involved in a protocol for a function? We will deal with three people, but k people would be similar. Let $f : \{0,1\}^n \times \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}$. One could define the complexity of f in terms of one person having x, one having y, and one having z and seeing how many bits are needed to compute $f(x,y,z)$. That is is not what is done. Instead they let Alice have y,z, Bob have x,z, and Carl have x,y.

Each player takes turns writing on a blackboard that the other two can see. With this model one can easily define the multiparty complexity of f. There are some applications of the lower bounds on multiplayer complexity to circuits (in Chapter 11). This work was not done with applications in mind, hence there are fewer of them.

In all the models looked at so far the string that Alice got and the string that Bob got were predetermined. What happens if we look at different ways to partition the input? One can look at best cases, worst cases, and cases where some of the input is available to both parties. The results obtained here are applied to VLSI (Chapter 8), decision tree complexity (Chapter 9), data structures (Chapter 9), branching programs (Chapter 12), and other topics. The work on variable partitions was done for the purpose of obtaining lower bounds on VLSI, hence its applications there are not surprising. Its applications elsewhere might be surprising.

3 Opinion

This is a great book! The topics are well chosen and the presentation is excellent. Each concept is illustrated with one or two well chosen examples. In fact, the examples are an integral part of the book.

The book mostly uses combinatorics, linear algebra, and the probabilistic method. Hence it can be read by a bright undergraduate. The authors provide a table (page xii) to answer the question 'If I want to read about topic XXX then which chapters do I need to read?' Hence if your main interest is (say) Decision Trees, then you need only read Chapters 7 and 9.

4. R_ϵ^{pub} is the number of bits needed in a protocol that allows a referee to flip coins that both Alice and Bob can look at. If the ϵ is unspecified then it is $\frac{1}{3}$.

5. Several other variants of randomized protocols are also defined.

In Chapters 1,2, and 3 (1) many techniques are developed to prove upper and lower bounds on $D(f)$, $N(f)$, $R_\epsilon(f)$, and R_ϵ^{pub} for a variety of f, and (2) many theorems comparing these classes to each other are given. Chapter 4 consists of advanced topics that draw on the knowledge of Chapters 1,2, and 3. We give a sampling of results from Chapters 1–4.

Upper and Lower Bounds

We define three functions below that will be referred to during this review. All of them are functions from $\{0,1\}^n \times \{0,1\}^n \to \{0,1\}$.

$$EQ(x,y) = \begin{cases} 1 & \text{if } x = y; \\ 0 & \text{if } x \neq y. \end{cases}$$

$$NE(x,y) = 1 - EQ(x,y).$$

1. $D(EQ) = D(NE) = n+1$.
2. $N(EQ) = n+1$. $N(NE) = 1 + \lceil \log n \rceil$.
3. $R(EQ) = O(\log n)$.
4. $R^{pub}(EQ) = O(1)$.

Theorems about Classes

For all the theorems below f is a function from $\{0,1\}^n \times \{0,1\}^n$ to $\{0,1\}$.

1. Let M be the $2^n \times 2^n$ matrix with rows and columns indexed by $\{0,1\}^n$. such that entry (x,y) is $f(x,y)$. Then $D(f) \geq rank(M)$. The proof involved linear algebra.
2. Let $g(x,y) = 1 - f(x,y)$. Then $D(f) = O(N(f)N(g))$. The proof is combinatorial.
3. $N(f) = \Omega(\log D(f))$. The proof is combinatorial. This is tight since $N(NE) = O(\log n)$ and $D(NE) = n$.
4. $R(f) = \Omega(\log D(f))$. The proof involves Alice and Bob actually exchanging bits that encode probabilities. This is tight since $R(EQ) = O(\log n)$ but $D(EQ) = n+1$.
5. $R_{\epsilon+\delta}(f) \leq R_\epsilon(f) + O(\log n + \log \frac{1}{\delta})$. This is proven by a probabilistic argument.

Advanced Topics

1. Assume $p \in N$. Let $f^p(x_1, y_1, \ldots, x_p, y_p) = (f(x_1,y_1), \ldots, f(x_p,y_p))$. Clearly $D(f^p) \leq pD(f)$. Can we ever obtain $D(f^p) << pD(f)$. The same can be asked for $N(f)$ and $R(f)$. Very few general theorems are known; however, it is known that $R(EQ^{\log n}) = O(\log n \log\log n) << (\log n)R(EQ) = O(\log^2 n)$.
2. One can define an analog to P in communication complexity by defining a 0-1 valued function f to be in P^{cc} if it can be computed with poly-log bits of communication. One can define NP and R similarly. Results from Chapters 1,2,3 show that $P^{cc} = NP^{cc} \cap coNP^{cc}$, $P^{cc} \neq NP^{cc}$, and $P^{cc} \neq coNP^{cc}$. In Chapter 4 they show that the set $DISJ(x,y)$ is $coNP^{cc}$-complete. ($DISJ(x,y) = 1$ iff x and y, as sets represented by bit vectors, are disjoint.)

Review of: **Communication Complexity** [1]
by Authors: Eyal Kushilevitz and Noam Nisan
Publisher: Cambridge University Press

Reviewed by W. Gasarch, University of Maryland at College Park

1 Overview

Assume that Alice has $x \in \{0,1\}^n$ and Bob has $y \in \{0,1\}^n$. They both want to know if $x = y$ but they want to exchange as few bits as possible. They can solve this problem with $n+1$ bits of communication (Alice sends Bob x, and Bob sends back 1 if $x = y$ and 0 otherwise.) The following questions come to mind

1. Can they do better than $n+1$ bits?
2. Can they do better if both of them can flip coins privately and they tolerate a small probability of error?
3. Can they do better if some referee flips coins publicly and they tolerate a small probability of error?

Communication complexity makes these questions (and many others) rigorous, and answers some of them. Many of the results are used to prove lower bounds on a variety of models such as VLSI, circuits, and decision trees. This is not an accident— much of communication complexity was developed with these goals in mind, and directly applied to them.

2 Summary of Contents

2.1 Chapters 1,2,3,4: Two Party Communication Complexity

Assume that Alice and Bob have unlimited computational power. Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$. Assume Alice has $x \in \{0,1\}^n$ and Bob has $y \in \{0,1\}^n$. Assume that, in all cases below, the goal is to have a protocol that ends with both Alice and Bob knowing $f(x,y)$. Our interest is in how many bits of information Alice and Bob need to exchange, hence refered to as 'the number of bits' Protocols are modeled by decision trees.

1. $D(f)$ is the number of bits required in a deterministic protocol to compute f.
2. $N(f)$ is the number of bits required for a nondeterministic protocol for f (every time a nondeterministic choice is made that will be charged as one bit of communication). If $f(x,y) = 1$ then some reachable leaf will have Alice and Bob both thinking that $f(x,y) = 1$. If $f(x,y) = 0$ then all reachable leaves will have Alice and Bob both thinking $f(x,y) = 0$. (The book uses a different but equivalent definition.)
3. $R_\epsilon(f)$ is the number of bits needed in a protocol that allows both Alice and Bob to flip coins, but also allows a probability of error of $\leq \epsilon$. If the ϵ is unspecified then it is $\frac{1}{3}$. (Any constant less than $\frac{1}{2}$ would suffice. The choice of constant only changes the multiplicative constant.)

[1] © W. Gasarch, 1997.