Joint Review of[1]
**Computational Complexity: A Conceptual Perspective**
**by Oded Goldreich**
**Published by Cambridge University Press, 2008**
**606 pages, Hardcover**
and
**Computational Complexity: A Modern Approach**
**by Sanjeev Arora and Boaz Barak**
**Published by Cambridge University Press, 2009**
**579 pages, Hardcover**

Review by
Daniel Apon dapon@uark.edu
University of Arkansas

# 1 Introduction

Both *Computational Complexity: A Conceptual Perspective* and *Computational Complexity: A Modern Approach* cover the field of computational complexity, which is a (if not *the*) central area in the theoretical foundations of computer science. Both are ultimately concerned with a single, fundamental question: How can we define a reasonable notion of *efficient computation* in the context of perennially limited resources?

The books are designed as textbooks for a course in complexity theory, aimed at early graduate students or even advanced undergraduate students. Every chapter is accompanied by a series of exercises with each book having hundreds of such problems for the reader to use for practice. There is no special prerequisite knowledge necessary to follow either book, aside from a reasonable degree of mathematical maturity, and as such, either could serve well in the role of a self-study text[2]. Additionally, due to their breadth of topics covered, both would be an exceptional reference text for experts in the field.

But naturally, the questions arise: How are they any different? How should I pick between the two for my own use? If I already own one, should I consider buying the other? And the answer is *obvious*: Get both! Of course, if this is not an option, then your answer is buried in the sub-title of each – specifically, in the distinction between the authors' choices of *conceptual perspective* and *modern approach*. This will be the driving theme of the review, and I will return to this idea as I review the two books.

# 2 Summary

## Computational Complexity: A Conceptual Perspective

Goldreich's book is divided into ten chapters, beginning with a discussion of models of computation and the history of complexity theory, progressing naturally through the various complexity classes,



[2]Disclaimer: I am originally self-taught in complexity from the Arora/Barak book. On the other hand, I'm not just being polite here: It *is* absolutely possible to use these books in a self-study!

and ending with more advanced concepts such as probabilistic proof systems, approximation, and average-case complexity. There is also a (quite hefty) set of appendices that comprise nearly a quarter of the pages of the text and that delves into some of the technical details and mathematical background omitted in the regular chapters. There is a strong, chapter-by-chapter, internal consistency to Goldreich's book, and in following his style, this summary of his book will proceed accordingly.

Chapter 1 begins with a high-level survey of the entire field of complexity. Beginning with a definition of Turing machines, this chapter builds up the historical and technical details to firmly place the rest of the book within a meaningful context. Of particular note, the first chapter spends some time to develop the intuitions underlying the theories of computability and complexity, highlighting the Church-Turing Thesis, the Post Correspondence Problem, and the Cobham-Edmonds Thesis. Additionally, there is discussion about uniform vs. non-uniform models of computation, oracle machines, machines that take advice, and a "sanity check" comparison between the Turing machine model of computation and more "real-world" models, such as an abstract RAM machine.

Chapter 2 develops the complexity classes **P**, **NP**, the **P** vs. **NP** question, and the theory of **NP**-completeness. There is a strong emphasis on the concepts of decision problems vs. search problems vs. promise problems. For example, Goldreich defines the search analogs of **P** and **NP**, respectively **PF** and **PC**. Additionally, *Cook-reductions*, *Karp-reductions*, and *Levin-reductions* are each separately developed here.

Chapter 3 is a shorter chapter entitled "Variations on P and NP." This chapter considers variations on the previous chapter's theme. Specifically, the non-uniform complexity class with advice **P/poly** is developed, as well as the Polynomial-time Hierarchy, **PH**. Examples are given of results that are motivated by a collapse of the Polynomial-time Hierarchy, e.g. $\mathbf{NP} \not\subseteq \mathbf{P/poly}$ unless **PH** collapses to its second level, and are related back to the **P** vs. **NP** question.

Chapter 4 formalizes the notion of various *hierarchy theorems*, e.g. there exist languages that can be decided in $O(n^2)$ time that cannot be decided in $O(n)$ time. In particular, the concepts of time- and space-constructible functions are explained, which lead into a discussion of the deterministic time hierarchy, the non-deterministic time hierarchy and the space hierarchy.

Chapter 5 builds an understanding of space complexity from the bottom up. Beginning with a comparison between time and space complexities, the chapter then jumps into a description of logarithmic space **L**, followed by non-deterministic space complexity, and ending with polynomial space **PSPACE** and a "game"-based discussion of the class.

Chapter 6 is devoted to topics in randomness and counting. First, the text considers the formulation of probabilistic complexity classes **BPP**, **RP** and **ZPP**. These classes are then related back to previously discussed classes, such as **P/poly** and the Polynomial-time Hierarchy – specifically $\mathbf{BPP} \subseteq \Sigma_2$. Then the discussion turns to exact and approximate counting problems, introducing the class **#P** and **#P**-completeness, and relates those concepts to **PH**, **NP** and **NP**-completeness.

Chapter 7, entitled "The Bright Side of Hardness," takes the complexity concepts that have been meticulously developed throughout the book thus far and provides useful applications for some of the more "negative" results. For example, we might not be able to efficiently invert one-way functions, but we can apply that fact in the design of cryptographic systems! Beginning with one-way functions, the chapter proceeds into an exposition of hard-core predicates, followed by average-case hardness arising out of worst-case hardness, Yao's XOR Lemma, list-decoding and hardness amplification.

Chapter 8 picks up from the previous chapter's concepts and develops a theory of pseudorandom

generators. In this chapter, the notion of computational indistinguishability is also formalized. A distinction is also made between general-purpose pseudorandom generators (useful for, say, cryptographic applications) and special-purpose pseudorandom generators that are tailored to tackle difficult computational complexity questions. For example, by the existence of a specific, space-bounded pseudorandom generator with exponential stretch in the space bound, we know that the log-space analog of **BPP**, **BPL**, is contained in **DSPACE**$(\log^2)$.

Chapter 9 surveys various types of probabilistic proof systems, specifically covering *interactive proofs* and the complexity hierarchy **IP**, *zero-knowledge proof systems* and the complexity class **ZK**, and *probabalistically checkable proofs* and the complexity hierarchy **PCP**. In particular, Chapter 9 describes the the proofs for **IP**=**PSPACE**, **IP**=**ZK** (under the existence of one-way functions) and the original proof of the PCP Theorem, **NP**=**PCP**$(\texttt{log}, O(1))$, as well as discussing *constraint satisfaction problems* (CSPs) and Dinur's later CSP-based proof of the PCP Theorem.

The final chapter, entitled "Relaxing the Requirements," begins with a short section on approximation and inapproximability. After mentioning the PCP-based inapproximability of a couple **NP**-complete problems and describing the notions of a *polynomial-time approximation scheme* (PTAS) and property testing, the chapter delves deeper into the topic of average-case complexity, where it describes a number of related analogs for traditional complexity classes, e.g. **distNP**, **sampNP** and **tpcBPP**.

As mentioned previously, the appendex section of *Computational Complexity: A Conceptual Perspective* is *very* extensive. Compared to the primary chapters of the text, the appendix comprises a series of largely self-contained discussions covering various topics in mathematics and special topics in complexity. In addition to a survey of necessary mathematical background for the book, subjects covered in the appendices include a survey of the quest for lower bounds, a theoretical discussion of the fundamentals of modern cryptography, advanced topics in probablistic complexity and randomization, and explicit constructions of error-correcting codes and expander graphs.

### Computational Complexity: A Modern Approach

Arora and Barak's book contains 23 chapters divided into three parts. The first 250 pages form "Part One: Basic Complexity Classes," the next 100 pages form "Part Two: Lower Bounds for Concrete Computational Models," and the final 200 pages form "Part Three: Advanced Topics." In addition, there is a short, 20-page appendix covering necessary mathematical background at the end of the textbook. As a result of this structure, and in contrast to Goldreich's book, the chapters of *Computational Complexity: A Modern Approach* tend to be more self-contained.[3] Therefore, it seems the most reasonable to view a summary of the current text in terms of the various *threads of thought* that bind disjoint chapters together, especially those that join introductory chapters from Part One with advanced chapters from Part Three.

The opening chapter of the book bears the (cheeky) title, "The computational model – and why it doesn't matter," and begins by formally defining Turing machines. Conceding that explicitly programming Turing machines is quite tedious, the chapter proceeds into a discussion of their expressive power (with respect to modern programming languages like Java or the C family), time-constructible functions, the deterministic time hierarchy, the Church-Turing thesis, the class **P**

[3]This will be elaborated on further in the Opinion segment of this review.

and Edmond's concept of "practical efficiency," i.e. that **P** encapsulates the notion of *efficient computation*.

Chapters 2 through 6 progress from the class **P** through the terrain of basic complexity classes. Chapter 2 includes a discussion of **NP**, **NP**-completeness, exponential-sized classes, Karp-reductions and Levin-reductions. (Cook-reductions appear in a Chapter 2 exercise.) There is a particularly interesting survey section on problems in **NP** that are not known to be **NP**-complete, such as factoring integers, unique games labeling and finding Nash equilibria. Chapter 3 covers diagonalization-based results, including Ladner's proof of the existence of "**NP**-intermediate" languages, and a discussion of oracle machines and relativization. Chapter 4 introduces space-constructible functions, the deterministic and non-deterministic space hierarches, as well as the space complexity classes **L**, **NL** and **PSPACE**. Chapter 5 continues by introducing the Polynomial-time Hierarchy **PH**, alternating Turing machines (ATMs) and a discussion of **PH** with respect to oracles. Finally, Chapter 6 wraps up the theme with a discussion of Boolean circuits and the class **P/poly**, including the development of the notion of uniform vs. non-uniform computation and ending with a discussion of circuit classes **NC** and **AC** as well as **P**-completeness.

Chapters 7, 8, 9 and 11 complete the introductory material in the book, beginning with the introduction of probabilistic Turing machines and progressing through a survey of the PCP Theorem. Chapter 7 discusses the basics of randomized computation including the classes **RP**, **coRP**, **ZPP** and **BPP**. Additionally, the topics of randomized reductions and randomized logspace algorithms are covered here. Chapter 8 introduces interactive proofs and the corresponding complexity classes **IP**, **MIP** and the Arthur-Merlin hierarchy **AM**[k], as well as a proof of **IP**=**PSPACE**. Chapter 9 is devoted to a survey of cryptography, which includes a discussion of negligible functions, one-way functions, pseudorandomness, psuedorandom generators and zero-knowledge proofs. Chapter 11 introduces the PCP Theorem (the proof is deferred until later) and demonstrates hardness of approximation results for `MAX-3SAT` and other problems. Additionally, there is a brief, introductory discussion of the Hadamard code.

Chapter 10 is worthy of special attention as it surveys quantum computation, which is unique to Arora and Barak's book. Topics covered include a formal definition of a qubit, quantum superposition, the EPR paradox, the complexity class **BQP** (with proofs of **BPP**$\subseteq$**BQP** and **BQP**$\subseteq$**PSPACE**), as well as detailed expositions of three famous quantum algorithms: Grover's search algorithm, Simon's period-finding algorithm and Shor's integer factorization algorithm.

Similar to Chapter 10, Chapters 12 through 16 (the chapters of Part Two) present material that, as a whole, is not found in Goldreich's book. Specifically, they cover *concrete models of computation* and the complexity lower bounds that result. Chapter 12 surveys decision trees and decision tree complexity, including Yao's Min-Max Lemma and the Boolean function characterizations of *sensitivity* and *block sensitivity*. Chapter 13 introduces concepts in communication complexity and various techniques for proving communication complexity lower bounds: the fooling set method, the tiling method, the rank method and the discrepancy method. Chapter 14 discusses circuit lower bounds, including Håstad's Switching Lemma, Razborov's Method of Approximations, the complexity class **ACC**0, monotone circuits and various barriers in circuit lower bounds[4]. Chapter 15 develops proof complexity and includes a discussion of propositional calculus, interpolation theorems, a proof of an exponential Resolution lower bound and further discussion of other such

[4]A meta-comment here: Arora and Barak do an excellent job of drawing connections between different lower bound techniques in concrete complexity. For example, Chapter 14's primary focus is circuit lower bounds but includes a short diversion into circuit lower bound approaches using communication complexity.

proof systems. Chapter 16 concludes the theme of concrete complexity by considering the restriction the basic operations of previous models, e.g. decision trees and Turing machines, to operations over a field or ring and topological techniques for proving lower bounds in such a setting.

Chapters 17 through 23 constitute Part Three, which focuses on advanced topics in complexity theory, and many of the chapters in this part directly build on concepts previously in the book. Chapter 17, building on the basic complexity classes, covers counting complexity, i.e. the class **#P**, including proofs of Valiant's Theorem (PERMANENT is **#P**-complete) and Toda's theorem ($\mathbf{PH} \subseteq \mathbf{P}^{\#SAT}$). Chapter 18 surveys Levin's theory of average-case complexity, introducing the notion of average-case reductions and the classes **distP** and **distNP**. Chapter 19, building on Chapter 18 and Chapter 9 (on cryptography), covers two related topics: hardness amplification and error-correcting codes. Topics in Chapter 19 include Yao's XOR Lemma, Impagliazzo's Hardcore Lemma, the Walsh-Hadamard, Reed-Solomon and Reed-Muller codes, and list decoding. Chapter 20, building on Chapter 7 (on randomized computing), discusses the notion of derandomization. Specific topics include derandomization from pseudorandom generators, the Nisan-Wigderson pseudorandom generator construction and the $\mathbf{BPP} \stackrel{?}{=} \mathbf{P}$ question. Chapter 21, building on concepts from at least Chapter 6, 7, 19 and 20, develops explicit constructions of various pseudorandom objects. Beginning with a discussion on random walks, eigenvalues and weakly random sources, the chapter proceeds through a construction of expander graphs and extractors, then develops a space-bounded pseudorandom generator. Chapter 22, entitled "Proofs of PCP theorems and the Fourier transform technique," picks up from Chapter 11 (on the PCP Theorem). Starting with a definition of CSPs, the chapter explicitly lays out Dinur's proof of the PCP Theorem, Raz's Parallel Repetition Theorem, Fourier analysis techniques for PCP proofs, the long code, an improved hardness of approximation result for `MAX-3SAT` and a similar result for `SET-COVER`. The chapter concludes with a discussion of the unique games conjecture and metric space embeddings. Part Three ends with Chapter 23, which builds on Chapter 14 (i.e. circuit lower bounds), by discussing the natural proof barrier to proving lower bounds.

# 3 Opinion

Let me begin with a relatively uninformative comment: Both books are excellent texts that thoroughly cover both the breadth and depth of computational complexity theory. "But *of course* that's the case," you say. "The authors of each are *giants* in theory of computing. I guessed as much with a glance at the cover of each book!" The interesting differences, then, come out of the *angle* from which each book approaches complexity theory. For instance, reading Goldreich's book, I get the unmistakable impression that I am on a step-by-step, start-to-finish journey through the landscape of complexity. Turning to Arora and Barak's book, however, I see a slightly wider array of subjects (*Quantum computation! Communication complexity! The unique games conjecture!*). There is a definite attempt in *A Modern Approach* to include very up-to-date material, while Goldreich focuses more on developing a contextual and historical foundation for each concept presented.

So, perhaps the largest difference in the books is in the style of presentation. Goldreich regularly pauses to offer up self-referential commentary or philosophical musings. There are numerous "teaching notes" embedded in each chapter where he opines on the best manner in which to present the material in a classroom setting. And in fact, it is impossible for a person to read *A Conceptual Perspective* from start to end without stopping periodically to reflect on the recurring question,

"So what does this all *mean* anyway?" By contrast, *A Modern Approach* sacrifices just a touch of the story-telling quality in the interest of presenting a wider range of subject material in a timely manner. Arora and Barak's book has a little bit more of a *lemma, lemma, then theorem, and let's get to the point* flavor, which is perhaps more reminiscent of the style of the majority of modern, conference publications. This is not to say Goldreich's book, by any means, lacks rigor (of which there is an abundance!) or that Arora and Barak's book ignores the context or history of various complexity results (each chapter ends with a discussion of the history of the topic in question). Rather, each book tends toward its own style, with Goldreich baking the narrative directly into the text and Arora and Barak pushing through the material and regrouping afterward.

Those intending to use either text in a course on complexity theory should be particularly aware of some of the more technical differences between the two books. Two specific examples from each book can help highlight these trends. The first is the difference in terminology used for the search analogs of **P** and **NP**. In Goldreich's book, they are called **PF** for *polynomial find* and **PC** for *polynomial check* and are presented nearly in tandum with their corresponding classes, whereas in Arora and Barak's book, there is mention of **FP** for *function polynomial time* in a later chapter. A search-**NP** is omitted in the latter, but would presumably be **FNP** for the sake of consistency. Perhaps this is only a minor quibble, but one should take care to ensure that students using either book are not confused if they later encounter different names for the same concepts or classes. The second example concerns the choice of notation used when presenting ideas. For instance, in the chapters presenting the concept of interactive proofs, Goldreich describes the strategy $f$ of each player in terms of $f(x, \gamma)$ for input $x$ and partial transcript $\gamma$, while Arora and Barak use a round-by-round notation – $a_1 = f(x), a_2 = g(x, a_1), a_3 = f(x, a_1, a_2)$, etc. Again, if the underlying concept is clearly taught, there should be absolutely no issue here. However, given that there are a litany of such examples between the two books, it may be useful for the instructor of a complexity course to be aware of both methods of explaining the concepts, and then pick and choose which they think fits best.

One final, major difference concerns the choice of topics included in each book. Goldreich is very up front in the introduction to *A Conceptual Perspective* in stating that the book only covers "high-level" ideas, by which he means technical material that, in his words, is clearly connected to "significant conceptual content." As a result, his book omits any mention of *concrete models of computation*, while Arora and Barak have an entire segment of their book (Part Two) devoted to topics in concrete complexity. Similarly, as mentioned before, *A Modern Perspective* has a chapter on quantum computation, while *A Conceptual Perspective* does not. On the other hand, this gives Goldreich more real estate to devote to more thorough explanations of the philosophy underlying the results, as well as to go slightly more in detail on special topics, such as the complexity foundations of cryptography.

In closing, if I were told to choose one book over the other, it would be an impossible decision. Both books have numerous, unique strengths and very few weaknesses. It really comes down to knowing which type of experience you are trying to find. However, theorists, researchers and instructors of any school of thought will find either book useful. I applaud all three authors for their outstanding contributions.