The Book Review Column[1]

by William Gasarch
Department of Computer Science
University of Maryland at College Park
College Park, MD, 20742
email: gasarch@cs.umd.edu

Welcome to the Book Reviews Column. We hope to bring you at least two reviews of books every month. In this column three books are reviewed.

1. **Cryptology and Computational Number Theory** by Carl Pomerance. Reviewed by Andreas Stein. This book is based on six lectures that Dr. Pomerance gave at an AMS meeting. It represents the state of the art in this field.

2. **Theories of Computability** by Nick Pippenger. Reviewed by Sanjay Gupata. This book is a textbook which covers regular languages, grammars, and computability theory.

3. **Vicious Circles** by Barwise and Moss. Reviewed by Alex Dekhtyar. This is a book that investigates what happens if you allow sets that are not well founded.

Review of: **Cryptology and Computational Number Theory** [2]
Editor: Carl Pomerance
Proceedings of Symposia in APPLIED MATHEMATICS
Published by American Mathematical Society in 1990
$44.00, Hardcover, 171 pages
ISBN number 0-821-80155-4
Reviewed by Andreas Stein, University of Waterloo
astein@math.uwaterloo.ca

1 Overview

Let n be an integer bigger than 1. Is n a prime? If n is composite, how difficult is it to determine its factorization? If one can factor n, then one can easily compute $\Phi(n)$, where Φ denotes Euler's totient function. Conversely, does there exist an easier way to compute $\Phi(n)$ than factoring n? These are the kind of questions which are of interest both for computational number theorists and for cryptographers. For instance, the most famous public key cryptosystem is RSA whose security depends on the presumed difficulty in factoring large composite integers.

The book is based on six lectures given at a short course on number theory and cryptology in August 1989 at the summer AMS meeting in Boulder, Colorado. Six chapters in the book, Chapter 2-7, correspond to the six presentations. Chapter 8 is an additional contribution which fits perfectly into this context, whereas Chapter 1 serves as an introduction to the topic. The contributions to the book discuss the most important subjects which are related to computational number theory and cryptology and can be seen as the state of the art at that time.

2 Summary of Contents

Chapter 1: Cryptology and Computational Number Theory- An Introduction, Carl Pomerance

Chapter 1 contains an elementary introduction to computational number theory and cryptology. Number theory provides most of the hard computational problems which can be used to guarantee the security of cryptographic schemes. It also provides solutions to some easy problems such as binary exponentiation (modulo a prime p). In this context, a definition of "easy" and "hard" computational problems is provided. The main efforts of the author in this chapter are to motivate most of the following chapters on a very basic level. He describes the principal problems which are related to primality testing (Chapter 2), factoring (Chapter 3), the discrete logarithm problem (Chapter 4) and knapsack cryptosystems (Chapter 5). Finally, the author focuses on the RSA cryptosystem whose security depends on the inability of quickly factoring composite integers of the form $n = pq$, where p and q are large primes. More precisely, if one is in possession of a method which quickly factors n, one can quickly compute $\Phi(n) = (p-1)(q-1)$, and, thus, break the RSA cryptosystem.

Chapter 2: Primality Testing, Arjen K. Lenstra

In Chapter 2, the author discusses the question how one can decide whether a number $n > 1$ is composite or prime. In his definition, *primality testing* means to provide a proof that the number n is prime. In more recent publications, a distinction between primality testing and primality proving is made. One defines *primality testing* to be a test for primality of a number n which, in case of primality, gives the result that either n is prime, or n is probably prime, or n is prime under some plausible heuristic assumptions. Whereas *primality proving* means to rigorously prove the primality of a number, of course, provided that it is in fact prime. The author summarizes the classical methods for primality testing and primality proving and then presents two powerful practical algorithms in detail. The first one is the *Jacobi sum test* which belongs to the class of *primality testing*. Hereby, the author describes the basic idea behind the test without going too deep into the mathematical background. The second method is the complex multiplication test which belongs to the class of *primality proving*. Basically, the complex multiplication test is a primality proof using ellipitc curves which proceeds with a DOWNRUN strategy.

Chapter 3: Factoring, Carl Pomerance

Chapter 3 explores the problem of factoring an integer n, i.e. finding its decomposition into a product of primes. Hereby, the author in principle discusses two practical methods for factoring large composite integers and explains the situations in which each of the methods has an advantage over the other. The first one is the quadratic sieve factoring method which the author classifies to belong to the "combination of congruences" camp. This method should be applied if n consists of only a few but large prime factors. For instance, if $n = pq$, where p and q are large prime factors, then this method should be used. The author then explains the multiple polynomial variation of the basic quadratic sieve method. This variation is of great practical importance, since it leads to a considerable speed up of the original method and allows parallel implementation. The second method is the elliptic curve factoring method which the author classifies to belong to the "groups of smooth order" camp. The method works best, if n has no large prime factors. The basic underlying idea of the elliptic curve factoring method is the same as the one of the $p-1$ factoring method which is also described in this chapter.

Chapter 4: The Discrete Logarithm Problem, Kevin S. McCurley

The *discrete logarithm problem* for a group G is defined as follows: Given $a, g \in G$, find an integer x such that $g^x = a$. By assuming that $G =< g >$, the author avoids the problem of

determining whether indeed $a \in < g >$ which might even be harder to solve. In this chapter, the author points out the cryptographic significance of this problem, since many cryptographic schemes rely on the presumed difficulty of the discrete logarithm problem in a special group and the inability of quickly solving this problem in certain large groups. The main subject of the paper is to discuss efficient algorithms for computing discrete logarithms. Hereby, he distinguishes between algorithms for general groups or *generic algorithms* which do not exploit any particular property of the underlying group structure and algorithms for groups with some additional knowledge of the group structure. In the first category, the most important ones are Shanks' baby-step giant-step method and Pollard's rho-method. If one knows that the order of the group G has no large prime factor, then the Silver-Pohlig-Hellman algorithm is quite feasible to compute discrete logarithms. If the group G possesses a special structure, then the probabilistic index calculus algorithm is most efficient.

Chapter 5: The Rise and Fall of Knapsack Cryptosystems, A. M. Odlyzko

In this chapter, the author discusses knapsack cryptosystems whose security is based on the difficulty in solving the *knapsack problem*, which is: given positive integers $a_1, \ldots, a_n$ and s, determine whether there is a subset of the a_j that sums to s. Various knapsack cryptosystems, for instance the Merkle-Hellman knapsack cryptosystem, have the advantage that they can be run at the high speeds. Unfortunately, most of these cryptosystems have been broken. The author then explains the Shamir attack to the basic Merkle-Hellman knapsack cryptosystem and describes a method to solving general low-density knapsacks which is due to Lagarias and Odlyzko.

Chapter 6: The Search for Provably Secure Cryptosystems, Shafi Goldwasser

This chapter is meant to be a description of cryptography as a formal science. The author discusses provable security of a cryptosystem against computational attacks. Expressions such as one-way functions, trapdoor functions, predicates and probabilistic polynomial-time algorithm are defined in view of computational complexity theory. Further topics include the formalization of secret-key and public-key encryption, zero-knowledge protocols and probabilistic encryption.

Chapter 7: Pseudorandom Number Generators in Cryptography and Number Theory, J. C. Lagarias

The author explains the importance of pseudorandom number generators for the use in cryptography and number theory. More exactly, he concentrates on the special case of pseudorandom bit generators. Intuitively speaking, these are functions that take as input a few random bits and deterministically produce as output a larger number of random-looking bits. The motivation is that pseudorandom number generators are a valuable computational resource, that computations should be allowed to be easily reproducible and that of constructing (provably) secure private key cryptosystems. The author provides a classification of number-theoretic generators and discusses the main ideas behind them. But, mainly, the author gives the formal definition of pseudorandom number generators in the abstract setting of the complexity theory. The basic result is that pseudorandom number generators exist if and only if one-way functions exist if and only if private key block cryptosystems can be constructed. Finally, three pseudorandom number generators are presented whose security relies on the computational difficulty of three number-theoretic problems.

Chapter 8: Odds and Ends from Cryptology and Computational Number Theory, Kevin S. McCurley

In this chapter, various applications of number theory in cryptology are described. Hereby, the author concentrates on those applications not discussed in previous chapters. He explains the computational difficulty of some number-theoretic problems and presents cryptosystems whose security depends on the difficulty of these problems. Mainly, the intention of the author is to stimulate further research in the area of computational number theory and cryptography.

3 Style

The book is a collection of papers written by several authors and therefore not intended as a textbook. As a consequence, the style of the chapters varies from elementary (for instance, chapter 1) to advanced (for instance, chapter 6). Nonetheless, the contributions are coherent and built up in such a way that there is not too much overlapping.

To understand chapter 1 one does not need any knowledge of the subject at all. For chapter 2-5 and 8, one just needs some basic knowledge in elementary number theory and linear algebra. In these chapters, the authors make every effort to keep the style as elementary as possible. Chapter 6 and 7 can be recommended to the advanced reader. This is mainly due to the intrinsic formalism involved with the complexity theoretic definitions.

4 Opinion

The book is an excellent introduction to the specified area and serves as well to freshen up the knowledge of the advanced reader. It was certainly the state of the art at the time it appeared.

I strongly recommend it as an introduction to cryptology and computational number theory, although one might have to bring it up-to-date. In particular, I like the efforts of all authors to motivate their special topic and to provide the reader with every possible insight. Furthermore, the references are exceptional and extensive.

Review of

Theories of Computability [3]

First Edition, 1997

Author: Nicholas Pippenger

Publisher: Cambridge University Press

Misc Info: $44.95, Hardcover, 251 pages, ISBN: 0-521-55380-6

Reviewer: Sanjay Gupta, Virginia Tech, sgupta@vt.edu

1 Overview

Having used Hopcroft and Ullman [HU79] during the undergraduate years, Lewis and Papadimitriou [LP81] during the graduate years, Machtey and Young [MY78] for the A.B.D. exam, Martin [Mar91] for teaching in the past, and having decided to teach using Sipser [Sip97] in the future, I was pleasantly surprised to discover that most of the presentation, if not the topics, in Pippenger's book were new to me. The first hint for this difference is in the title "**Theories** of Computability," as against some variant of "**Theory** of computation" in the other books. The book presents several alternate formulations of the theory of finite state machines, grammars, and computable functions.

- Chapter 1 introduces the basic mathematical objects used throughout the book.

- Chapter 2 discusses finite automata and their languages.
- Chapter 3 discusses grammars and their languages.
- Chapter 4 discusses computable functions and relations.

2 Summary of Contents

2.1 Chapter 1

The chapter starts with some interesting mathematical stories to help define the author's intuitive notion of computability, which affected his choice of topics covered in the book. As the author states "Indeed, we might go as far as to say that computability is the study of the consequences of imposing finiteness conditions. Of course, we cannot discover the essence of computability in such a superficial slogan ...."

The role of finiteness conditions, in terms of a finite set of axioms for various mathematical objects and the properties following from these conditions, is indeed the unifying theme among a variety of topics presented in the book.

The main mathematical object defined in this chapter is *clones*. A class $\mathcal{P}$ of boolean functions is a clone if it satisfies the following conditions:

- For each $n \geq m \geq 1$, $\mathcal{P}$ contains the projection functions: $\text{proj}_{n,m}(x_1, \ldots, x_n) = x_m$.
- For each $n, m \geq 1$, each n-ary function $f \in \mathcal{P}$, and m-ary functions $g_1, g_2, \ldots, g_n \in \mathcal{P}$, $\mathcal{P}$ contains the composite function: $h(x_1, \ldots, x_m) = f(g_1(x_1, \ldots, x_m), \ldots, g_n(x_1, \ldots, x_m))$.

The chapter describes Post's lattice of all boolean clones without proof. However, the following important consequence of Post's lattice is proved: Every boolean clone has a finite basis. Thus, every boolean clone is generated by a finite set of boolean functions. For example, the smallest boolean clone consisting of only projection functions can be generated by the empty set and the largest boolean clone consisting of all boolean functions can be generated by $\text{NAND}(x_1, x_2)$.

The chapter also describes boolean relations, co-clones, and includes a section on the prospects of generalizing the theory of boolean functions and relations to domains with more than two elements. A proof of the Galois correspondence between the lattice of clones and the lattice of co-clones is included.

2.2 Chapter 2

This chapter starts with a description of finite state machines without using any explicit machines. Instead, Nerode's notion of *intrinsic state* is used.

Let A be any set, $A^* = \cup_{n\geq 0} A^n$, and $A^+ = \cup_{n\geq 1} A^n$. Given a function $f : A^+ \to A$, define $(x^{-1}f)(y) = f(x \cdot y)$.

$(x^{-1}f)(y)$ is called the intrinsic state reached by f under x. Also, f is finite state if the set $\{x^{-1}f : x \in A^*\}$ of intrinsic states is finite.

Languages over an alphabet A are introduced as subsets of the free monoid A^*, where the operation is concatenation and the identity element is ε, the empty string. Closure properties of languages which can be "recognized" by finite state machines are studied. These languages are defined to be *recognizable* languages.

The regular expressions are defined and shown to be equivalent to recognizable languages. Instead of using Kleene's proof, the author presents Eilenberg's proof which has the merit of producing a canonical regular expression. The regular expression produced by Kleene's proof is highly dependent on the initial numbering of states.

Recognizable languages are also shown to be equivalent to closed logical expressions using quantifiers ($\exists$,$\forall$) over natural numbers and sets of natural numbers. An interesting consequence of this result is that every closed logical formula is equivalent to one in which all set variables are existentially quantified by the outermost quantifiers of the expression.

The following sections in the chapter present a one-to-one correspondence between "varieties" of recognizable languages and "varieties" of finite monoids, starting with the example provided by the variety of aperiodic languages. Also included are descriptions of some variations of finite automata.

2.3 Chapter 3

This chapter starts with the standard treatment of regular languages, linear languages, context-free languages, and (recursively) enumerable languages, using various restrictions on generative grammars. The sections on context-free languages include Chomsky normal form, Greibach normal form, and ambiguous languages. A specific language is shown to be inherently ambiguous.

The section on enumerable languages includes several examples of languages which are enumerable and an example of a language which is not enumerable, introducing the diagonalization argument.

After the standard treatment, the families of languages mentioned above are put in the same general axiomatic framework using *rational cones*. A family of languages $\mathcal{C}$ is a rational cone if it satisfies the following conditions.

- $\mathcal{C}$ is closed under homomorphic reflections: if $L \in \mathcal{C}, L \subseteq B^*$ and $h : A^* \to B^*$ is a homomorphism, then $h^{-1}(L) \in \mathcal{C}$.
- $\mathcal{C}$ is closed under intersections with regular languages: if $L, R \in A^*, L \in \mathcal{C}$ and R is regular, then $R \cap L \in \mathcal{C}$.
- $\mathcal{C}$ is closed under homomorphic images: if $L \in \mathcal{C}$, $L \subseteq A^*$ and $h : A^* \to B^*$ is a homomorphism, then $h(L) \in \mathcal{C}$.

A rational cone is called *principal* if it is generated by a single language. Linear languages, context-free languages, and enumerable languages are all shown to be principal rational cones generated by specific languages.

The last section in the chapter describes a connection between languages and standard generating functions, where the coefficient of the term ξ^k in the generating function corresponds to the number of words of length k in the language. Though the structure of the language is lost in this representation, some interesting results about the structure of the generating functions corresponding to regular and context-free languages are proven. These results nicely blend with the author's notion of computability defined using the mathematical stories in the first chapter.

2.4 Chapter 4

The chapter starts with the axiomatic description of a class of functions $\mathcal{P}$ called *reflexive class*. Without using any machine description, it is shown that the famous halting problem is not in $\mathcal{P}$. Rice's theorem on index sets and Kleene's recursion theorem are also proven for the reflexive class.

Subsequently, a simple machine model *register machines* is introduced and the class of functions computed by the register machines are shown to form the smallest reflexive class. This smallest reflexive class is also shown to be equivalent to *recursive clones*, a class of functions generated by projections, compositions, zero function, successor function, primitive recursion, and the minimalization operator (partial recursive functions). Of course, instead of register machines any "reasonable" machine model would have sufficed.

All the above definitions are nicely tied together using oracles in the register machines. Any reflexive class is the class of functions computed by register machines relative to some oracle. The results on functions are generalized to sets using the characteristic function of a given relation. Also included is Friedberg's enumeration without repetition of all partial recursive functions.

The following sections include several advanced topics on register machines relative to oracles. The concepts of Turing-reducibility, T-degree, and T-completeness are introduced and Friedberg and Muchnik's construction (using the priority argument) of two incomparable sets, with respect to Turing-reducibility, is given. This result is applied to a seemingly unrelated scenario to prove Ramsey's theorem that every infinite graph has either an infinite clique or an infinite anti-clique.

Subsequently, notions of many-one reducibility, creative sets, and one-one reducibility are introduced and Myhill's result that all creative sets are recursively isomorphic is proved. All universal functions are also shown to be recursively isomorphic.

The final sections include brief introductions to abstract complexity theory (descriptive complexity measures and computational complexity measures with applications in prediction and inductive inference) and computable real numbers.

3 Style

The preface clearly defines the character of the book. "The methods of the book are mathematical" and "...the book is written for those making their way to the frontier of research."

The style in general is terse and to the point, though on almost all occasions the author explains the intuition behind the definitions and results. The definitions are usually part of the descriptive text; the theorems are marked clearly using itemized text. The book should be read sequentially and is not meant to be a reference book. The problems are marked E,M,H,U, denoting, easy, moderate, hard, and unsolved respectively. Many tangential results are mentioned without proofs, but the author has been very thorough in attributing results to the literature.

4 Opinion

This is an excellent textbook for a beginning researcher who wants to familiarize himself/herself with several axiomatic frameworks for the theory of computability. It is not suitable for undergraduates or even beginning graduate students in computer science, unless they have a good background in logic. I enjoyed the nuggets of wit spread throughout the book, providing a pleasant contrast to the abstract material. For example, a proof involving a two person game is in "an anthropomorphized form" and the alphabet is quadrupled by coloring letters pink, blue, both (violet), none (white), because "color leads to more vivid description of constructions."

References

[HU79] J.E. Hopcroft and J.D. Ullman. *Introduction to Automata theory, languages, and computation.* Addison Wesley, 1979.

[LP81] H.R. Lewis and C.H. Papadimitriou. *Elements of the theory of computing.* Prentice-Hall, Inc., 1981.

[Mar91] J.C. Martin. *Introduction to languages and the theory of computation.* McGraw-Hill, Inc., 1991.

[MY78] M. Michael Machtey and P. Young. *An introduction to the general theory of algorithms.* Elsevier North Holland, Inc., 1978.

[Sip97] M. Sipser. *Introduction to the theory of computation.* PWS Publishing Company, 1997.

Review of *Vicious Circles* by
Jon Barwise and Lawrence Moss
Published by Cambridge University Press in 1996
Softcover,$24.95, 390 pages
ISBN number 1-575-86008-2

Review by
Alexander Dekhtyar
University of Maryland
dekhtyar@cs.umd.edu

1 Overview

Since its very first days, modern set theory had "paradox" written all over it. For the past hundred years some of the best mathematitians of late 19th and 20th century (such as Cantor, Russel and Tarski) would devote their attention to the analysis of set theory, paradoxes it creates and attempts to modify it in order to escape these paradoxes.

As a result of the attempts to rid the set theory of the class of paradoxes (usually associated with Bertrand Russel's name), which appeared when one would start to look at sets that can be elements of themselves, Russel proposed a set theory which would iteratively build all sets in such a manner that self-inclusion became impossible.

This would have sufficed if it wasn't for the fact that quite a number of events and objects, which present interest to scientists exibit circular or repetative behaviour, and / or are self-referential. And in many cases it turns out that Russel's set theory is too prohibitive to model these phenomena.

In the book under review, Jon Barwise and Lawrence Moss set a goal of providing a framework for constructing models for such circular events. The book starts with a brief introduction into set theory. After that the reader is provied with a set of examples of circularity as it is prohibited by set theory from various backgrounds. After that the basic theory of circularity, based on set theory is introduced and its applications are discussed. Finally the theory is enhanced in later chapters and at the end of the book the complex applications are being described.

The more detailed summary of the book is provided below.

2 Summary of Contents

The book is broken into parts, which in turm are broken into chapters.

Part I of the book, **Background**, contains two chapters, *Introduction* and *Background on set theory.* The former chapter contains a brief description of the problem domain and explains the contents of the book. The latter chapter is devoted to brief introduction into the set theory.

The authors describe set theory axioms. The author's approach to the discussion of the axioms is motivated by the goal of establishing the framework for modelling circularity in further chapters. It is shown exactly how and to which extent the axioms prohibit circularity. As a core "nemesis" for circularity, Foundation Axiom is introduced and its implications are discussed.

In **Part II** of the book, **Vicious Circles**, the problem of circularity is introduced. The authors provide a number of examples of circularity in Computer Science (*chapter 3*), philosophy, linguistics and mathematics (*chapter 4*). In the heart of most of the examples is an attempt to operate with a structure that has the form $x = \langle a, x \rangle$, or $x = \{\{a\}, \{a, x\}\}$. It is easy to see that this construction has itself a part and therefore if Foundation Axiom is accepted, it is not a set.

The last chapter of this part, *Circularity and paradox*, is devoted to establishing the link between the well-known set theory paradoxes (such as the Liar Paradox and Russel's paradox) and circularity as it is studied in the book.

Part III, Basic Theory is the core of the book. In *Chapter 6, The Solution Lemma* the alternative to Foundation Axiom is introduced. The first formulation of this alternative, called Anti-Foundation Lemma is due to Forti and Honsell (1983). Among the number of ways to present Anti-Foundation Axiom the authors choose one called Solution Lemma, based on the following approach. The notion of flat system of equations is introduced. Equations considered are of a form similar to described above: $x = \{a, x\}$. The Solution Lemma formulation of the Anti-Foundation Axiom is then defined as a postulate that every flat system of equations has a *unique* solution. Now, constructions like the one above or $x = \{x\}$ become the representations of valid sets in the set theory obtained by replacing the Foundation Axiom with Anti-Foundation Axiom (the axiom system obtained by this will be referenced further as ZFA).

Chapter 7, Bisimulation deals with the question of equivalence of newly defined sets. In which cases can we say that two flat systems of equations as defined in *Chapter 6* have the same solution? This question is answered by introduction of a notion of bisimilation. The main theorem says that two flat systems of equations have the same solution sets iff they are bisimilar. It is shown that bisimilarity is a true equvalence relation (i.e. it is reflexive, symmetric and transitive). Later in the chapter the notion of bisimilation is extended from systems of equations onto sets, which in turn solves the purpose of establishing strong extentionality of newly defined sets. The chapter ends with discussion on how to computer the bisimilation relation for two systems of equations effecitively.

In *Chapter 8, Substitution* the notions of system of equations, solution sets and Solution Lemma introduced in *Chapter 6* are refined and an alternative formulations for Anti-Foundation Axiom are provided. The main motivation for seeking refinement of the notion of system of equation is the fact that although it provides for very simple formulation of Anti-Foundation Axiom, the application of this formulation is rather difficult due to the restrictive nature of the notion of flat system of equations.

In this chapter the general systems of equations which have more complex syntax are introduced. The chapter then proceeds to define the solution set for a general system of equations. The General Form of the Solution Lemma is presented at the end of the chapter.

Chapter 9 is a standalone chapter in which the authors show that the new set theory (with Anti-Foundation axiom) is an extention of the set theory with Foundation Axiom. Also, relative consistency of the new set theory is established.

Part IV, Elementary Applications seeks to establish the connection between the theory described in **Part III** and a variety of notions from Graph Theory, Logic, Game Theory and other disciplines.

In *Chapter 10, Graphs*, yet another version of Anti-Foundation Axiom, based on graph theory is constructed. The key notion, defined in the chapter is a decoration of a graph G. Decoration is

such a function d from nodes of G to sets, that

$$d(a) = \{d(b) | (a, b) \text{ is an edge in } G\}$$

After the properties of graph decorations are studied, the formulation of Anti-Foundation Axiom as "Every graph G has a unique decoration" is given. The chapter also translates the notion of bisimilarity onto graphs (two graphs are bisimilar iff they have the same decoration).

Chapter 11 establishes the connection of the new set theory with modal logic. Language $\mathcal{L}_\infty(A)$ is defined (as an extention of standard modal logic longuage $\mathcal{L}(A)$ with conjunction over (possibly infinite) set of formulas). Semantics of this language is given in terms of Kripke structures. The connection between the validity of formulas $\mathcal{L}_\infty(A)$ on a Kripke structure G and its decoration (as Kripke structure is a labeled graph, it has a decoration as defined in previous chapter) is then established and bisimilar Kripke structures are studied.

Second part of this chapter is devoted to characterizations of sets defined in ZFA. A formula Θ (of $\mathcal{L}_\infty(A)$ or $\mathcal{L}(A)$) is said to *characterize* a set a iff Θ is valid only on a. The main result of this part of the chapter is that every set (defined in ZFA) is characterizable by some sentence in $\mathcal{L}_\infty(A)$ or $\mathcal{L}(A)$. Proof of this result however is not simple, and most of rest of the chapter is devoted to it. In the last part of the chapter different axiomatizations of modal logic are given[4].

Capter 12, Games starts with a brief introduction to game theory. A definition of a 2-player game is given and some properties of such games are established. Then, the correspondence between games and the validity of first-order formulas is discussed (Ehrenfeucht-Fra$\ddot{i}$sse or "pebble" games).

With this information as background, the authors proceed to describe a "pebble game" for determining whether two sets are bisimilar[5]. Then the resolution of *HYPERGAME*, one of the paradoxes presented in *Chapter 5*, is given.

Chapter 13 deals with resolutions of the rest of the paradoxes, described in *Chapter 5*. The main target is the Liar paradox. To approach it, first three valued logic and partial models are introduced. Then, the Liar paradox is studied in the established framework.

Part IV ends with *Chapter 14* devoted to streams. Strams were first introduced, among other examples in *Chapter 3*. As stream is a pair, whose first element is an "atom" and second element is a stream ($s = \langle a, s \rangle$). This is very similar to the notion of *lists* with the sole difference being that lists are finite, while streams are defined to be infinite. In this chapter, a closer look at streams is taken. More than in establishing the actual properties of the streams, however, the authors are interested in developing and applying the new methods (coinduction, corecursion) and demonstrating how these methods can be used in proofs.

Part V of the book consists of three chapters which contain some more theory.

Chapter 15 contains an overview of fixed-point theory. The authors define notions of fixed point, least fixed point and greatest fixed point for monotonic operations on sets. Theorems about the existance of both least fixed points and greatest fixed points for monotonic operators are stated and proven and the properties of both are studied. Last but not least, a connection between fixed points and games is shown at the end of the chapter.

Chapter 16 ties together the notions of a flat system of equations and of greatest fixed points. First, it is shown how a flat system of a equtions can be treated as a Γ-coalgebra $\langle X, e \rangle$ where X is a set and $e : X \leftarrow \Gamma(X)$ for some operator Γ. The notion of a Γ-morphism between the two Γ-coalgebras is introduced. Then the notion of solution of a flat system of equations is extended on coalgebras and a structure of a solution set for coalgebras for monotonic operators which map sets onto pure sets (called proper operators) is described.

The authors then introduce uniform operators as monotonic, proper operators that have one additional property: they commute with almost all substitutions. The main property of uniform

operators Γ is that their greatest fixed point is exactly the union of solution sets of all *Gamma*-coalgebras. Finally, if Γ is a uniform operator, then any Γ-coalgebra has a *unique* soultion, which is a subset of the greatest fixed point of Γ.

Chapter 17 is devoted to corecursion. While recursive definitions define mappings from sets to their least-fixed points (and are very well understood and widely studied), the defintions of mappings from sets to their greatest-fixed points (i.e. corecursive definitions) are far less well understood. In the chapter, the notion of uniform operator is strengthened (the new class of operators is called smooth operators), and the notion of corecursion relative to these operators is studied.

Last part of the book (**Part VI**) consists of three chapters that provide more applications of the theory devloped in **Part III** and **Pert V** and a chapter that contains authors' conclusions.

In *Chapter 18* the greatest fixed points of some important operators are studied. *Chapter 19* describes modal logics associated with the operators described in *Chapter 18*.

Chapter 20 contains a detailed philosophical discussion of the ZFA system of axioms (and other related systems of axioms). Finally in the last chapter of the book (*Chapter 21*) the past, the present and future of the area are discussed. Together with a short history of the field, the authors present an extensive set of open problems.

3 Style

From a book on foundations of mathematics (and we can say that **Vicious Circles** is about foundations of mathematics) one should expect very precise and formal presentation of material. One the other hand, a book that seeks to extend upon the foundations of mathematics, and provides formal presentation can easily become unreadable.

In this book the authors make a solid attempt to combine both formality of presentation and clarity of explanations. And for most part (except for rather complicated content of **Part V**) this attempt succeeded. While the notation used in the book is rather diverse and sometimes complex, and while the definitions, statements of theorems and proofs are formalized, authors make every attempt to explain what is going on in plain English. The book contains plenty of examples, that cover virtually everything.

Those who want to test their understanding of the material presented in the book, will be able to do so by doing many exercises found in the book. Some of the exercises are similar to the examples given, while other require deep understanding of the material and proof techniques to be used and are quite challenging. The answers to all exercises are given in the end of the book.

It would be fair to say that the authors also succeeded in their desire to make the book mostly self-contained. As it is seen from the descriptions of various chapters of the book, wherever information from a certain area of mathematics were to be used, this information had been included (f.e. set theory, modal logic, three valued logic etc.). So it is indeed possible for a reader with only general mathematical background to read this book without getting lost. However, because of the plentitude of different areas of mathematics, and logic in particular brought up in the book, extra knowledge in these or similar areas is certainly more than helpful.

Last, but not least, the diagram of the content dependencies between the chapters of the book is very useful, esp. if the book is to be used as a reference or a textbook for a graduate course.

4 Opinion

The Anti-Foundation Axiom, which lies in the core of the theory described in the book was first introduced in 1983, i.e., less than 20 years ago. This fact makes *Vicious Circles* a somewhat unique book: on one hand it is devoted to the very foundations of mathematics, while on the other hand, the material presented is *relatively* new.

As such, this book is a valuable collection of information about the theory of circularity and its relation to other well-established areas of mathematics.