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 four books are reviewed.

1. **Stable Marriage and its Relation to Other Combinatorial Problems: An Introduction to Algorithm Analysis** by Donald Knuth. Reviewed by Tim McNichol. This book uses the stable marriage problem as motivation to look at some mathematics of interest. It would be useful for undergrads; however, for a serious study of matching there are more advanced and more up-to-date books available.
2. **The Limits of Mathematics** by Gregory J. Chaitin. Reviewed by Vladimir Tasic. This book is on algorithmic information theory and randomness as they relate to Berry's Paradox ("the shortest number that requires less than 1000 characters to describe it" has just been described by that phrase in quotes, yet that phrase was less than 1000 characters.)
3. **Privacy on the Line** by Whitfield Diffie and Susan Landau. Reviewed by Joseph Maklevitch. This book is about the balance between the citizen's need for privacy and the government's need to intrude to prevent or solve crimes. These issues are relevant now because of crytography and computers. The authors are respected theorists who have worked in cryptography, hence their comments are worthy considering. This book has caused some controversy in the math community— see the June-July 1998 issue of *Notices of the AMS*, also available at http://www.ams.org/notices. Or, better yet, read the book!
4. **A Theory of Objects** by Authors: Martín Abadi and Luca Cardelli. Reviewed by Brian Postow. This book is about formalizing the semantics of object-oriented languages. To do this, a new calculus is introduced.

Review of
Stable Marriage and its Relation to Other Combinatorial Problems:
An Introduction to Algorithm Analysis[2]
Author: Donald E. Knuth
Published by American Mathematical Society in 1996
$19.00, Softcover
ISBN number 0-821-80603-3

Reviewer: Timothy H. McNicholl

1 Overview

This book is a translation of the revised and corrected edition of *Marriages stables et leurs relations avec d'autres problèmes combinatories* published in 1976 by *Les Presses de l'Université de Montréal* . It is based on a series of lectures given by the author at the Centre de recherches mathematicques.

[1] © William Gasarch, 1999.
[2] ©1999, Tim McNichol

The purpose of the book is to give an informal introduction to algorithm analysis in which the main ideas are introduced by example rather than by theorems.

The chief example used is the stable marriage problem which can be defined as follows. A finite set of 'men' is given along with a set of 'women' of the same cardinality. In addition, each person is assumed to have a ranking of the members of the opposite sex. A *matching* is a bijection between the two sets. We think of a matching as a set of n monogamous marriages. An *unstable* matching is one in which there is a man X and a woman Y such that X ranks Y higher than his spouse and Y ranks X higher than her spouse. The methods used to find the mean number of steps required by this algorithm are then applied to an algorthm to find the shortest distances between a node and all the other nodes in a graph. An algorithm for storing information in a table by hashing is then considered and the results obtained are used to develop a lower bound on this mean in the case where the women all have the same ranking of the men. The asymptotic value of this mean as a function of the number of men is then obtained using a probabalistic method.

2 Summary of Contents

The material is presented in seven 'lectures'. The first of these defines the stable marriage problem in a suitably informal style (no mention is made of linear orders and the like; rather, rankings are presented via matrices). Examples are given to illustrate stable and unstable matchings as well as to demonstrate that many 'societies' have several stable matchings. This raises the issues as to whether some stable matchings are better than others and whether there exists an 'optimal' stable matching. These issues are addressed in the problem set for this chapter via the following provocative exercise: show that the matching obtained by marrying each man to the highest ranked of all brides he can obtain in *some* stable matching is a stable matching. This matching, which is 'male-optimal' by definition, turns out to be 'female-minimal' in the sense that no woman can do any worse than she does in this matching.

Lecture 2 presents an algorithm for obtaining a stable matching. Although this terminology is not used in the text, the algorithm could be called the 'courtship algorithm' since it mimics how marriages are formed in many societies. The algorithm is mercilessly presented in pseudo-ALGOL, but its operation can be summarized as follows. Each man continues to make marriage proposals until he becomes permanently engaged. Only one man is allowed to make proposals at any point in time, and whenever a man makes a proposal, he does so to the woman he ranks highest out of all those who have not previously rejected or 'dumped' him. A woman rejects a proposal only if she is already engaged to a man she likes better than the suitor; otherwise she dumps her current fiance in order to accept the proposal. It is shown that the algorithm eventually terminates and that when it does the matching obtained is stable. Furthermore, it is shown that the matching obtained is the male-optimal matching discussed in the exercises for lecture 1. An example is laborously worked thorugh in order to illustrate the operation of the algorithm. More insight into the algorithm's machinations could have been conveyed if an informal presentation of the main ideas involved had preceeded the pseudo-code.

The number of propsosals made during the course of the algorithm provides a good estimate of the number of steps needed by the algorithm, and the former quantity is analyzed in lecture 3. It is shown that the mean number of proposals made when there are n men and n women and the preference matrix of the women is fixed is no more than $(n-1)H_n + 1$ where H_n is the sum of the first n terms of the harmonic series. Only an upper bound is obtained since the problem is simplified by assuming that the men have 'partial amnesia'. Namely, whenver they make a proposal, they can only remember the last woman who has dumped or rejected them whenever they are making

a proposal. As a result each man possibly makes some redundant proposals and hence and upper bound is obtained. The assumption of partial amnesia allows the problem to be almost completely reduced to that of determining the probability that a coupon collector who already posseses m of n coupons next obtains a coupon already in her collection.

In the fourth lecture, these techniques are applied to Dijkstra's algorithm for finding the shortest distances from a node in a graph to each of the other nodes in the graph. The application is made via an analogy between the nodes in the graph and the women in the marriage algorithm. The logic is glossed over, but it is demonstrated that the mean number of steps in Dijkstra's algorithm is no more than the mean number of proposals in the marriage algorithm.

The fifth lecture considers the searching of a table via hashing. It is shown in a straightforward manner that when there are m items in a table with n storage blocks, then the mean number of steps required to place the next item is

$$\frac{n+1}{n+1-m}.$$

This result is then used to obtain the mean number of steps in the marriage algorithm in the special case where all the women have the same ranking of the men. It is shown that in this case the mean number of proposals when there are n men is $(n+1)H_n - n$. It is conjectured that this is a lower bound on the mean number of proposals when the preference matrices for both sexes are arbitrary. The asymptotic value of this mean is then obtained using a probabilistic approach and is shown to be $nH_n + O(\log^4 n)$.

The sixth lecture gives an exact implementation of the marriage algorithm in ALGOL along with a recursive algorithm for finding all stable marriages for given sets of men and women and associated preference matrices. Lecture 7 presents a list of open problems some of which have been solved since the book was first published and some of which have not.

3 Opinion

The text provides many interesting examples and problems for students to study while learning the topic of algorithm analysis. However, the writing is too terse at some key points for this book to be used as the main text for a course in this subject. Perhaps this is due to the murkiness endowed by repeated translation (the lectures were given in english to french students; they were later translated into french and a translation into english is the subject of this review). But a professor willing to fill in some of the gaps could use this book as a supplementary text.

A more modern book on the subject which the serious resercher might want to consult is *The Stable Marriage Problem* by Gusfield and Irving (MIT Press, 1989).

Review of
The Limits of Mathematics[3]
Author: G.J. Chaitin
Published by Springer Verlag in 1998
Hardcover, $32.00
160 pages
ISBN 981-308-359X

Review by
Vladimir Tasic
University of New Brunswick
vlad@conway.math.unb.ca

If you find pleasure in being baffled by the austerity of logicist incompleteness proofs based on Berry's paradox, this is not a book for you; Boolos's minimalist gem (which appeared in the "Notices of the A.M.S." a few years ago) is your natural choice. If, on the other hand, you actually want to learn something about the relationship between Berry's paradox, randomness and incomplete-ness pheno-mena, I recommend"The Limits of Mathematics". Chaitin has invested considerable energy into explaining his way of thinking about the topic, from the point of view of algorithmic information theory. This book is primarily concerned with the "why" and the "how" of limitative results. The ideas are carefully motivated, revisited and reinforced throughout, emphasizing intuitive understanding rather than a dryly formal"theorem-proof" approach. The result is a book that leaves the reader with the feeling of having witnessed one of those rare events: a good lecture. "The Limits of Mathematics" is not intended to be bed-time reading. It requires active participation of the reader, who is challenged to supply the details and invited to try out the software that comes along with this course. Admittedly, the fifty pages of code at the end of the book might appear slightly intimidating to those of us who quit programming upon encountering COBOL. However, the presentation of algorithmic information theory in terms of an explicit complexity measure based on a modified version of LISP is one of the key features of the book. In addition to making possible the hands-on approach which the author suggests, dealing with a suitably chosen LISP dialect allows Chaitin to establish explicitly some of the constants that occur in complexity estimates. For example, it is derived that the complexity (in Chaitin's sense) of the bit-string consisting of the first N bits of the halting probability must be greater than $N - 8000$. Various other results are made explicit, including the bound on the complexity of the theorems of a formal axiomatic system. This incompleteness theorem is used to make the case for a "quasi-empirical" philosophy of mathematics and the use of computers as tools for mathematical experimentation. I am not a specialist on algorithmic information theory; having read this book, I feel I understand something about this field.

Review of
Privacy on the Line[4]
Author: Whitfield Diffie and Susan Landau
Published by MIT Press 1998
Hardcover, $25.00
360 pages
ISBN 0-262-04167-7

Review by

Joseph Maklevitch
York College (CUNY)
joeyc@cunyvm.cuny.edu

In his book A Mathematician's Apology, the pacifist G.H. Hardy attempted to take comfort from the fact that number theory, his area of specialty, might never be put to any use, especially use that Hardy would not have approved of. Hardy was naive. Stanislas Ulam, the "pure" mathematician turned physicist, is credited with being the co-inventer of the hydrogen bomb - a dubious honor. Yet the path that ideas will lead to, even those initially developed for reasons of unlikely value to mankind, are difficult to chart. What is perhaps closer to the truth than Hardy's hope is a thought of the topologist Leo Zippin: if some mathematician is clever enough to find what seems to be hopelessly abstract results, some other mathematician will be clever enough to find a use for the results. Many theorists in mathematics and computer science today live in a world more defined by the potential of big dollar signs than that their work may sit admired merely for its "beauty and elegance" in a scholarly journal. Today, theoreticians may be reluctant to publish some of their "beautiful" work in a scholarly journal too soon, lest this action serve as an impediment to the use of the idea as part of a patent application. Software patents and other emerging trends in intellectual property law are becoming part of the ivory tower world. Hardy, I suspect, would not have approved.

Although the issue of the good and evil that can be a consequence of one's work in mathematics or computer science is not explicit in the new book Privacy on the Line by Whitfield Diffie and Susan Landau, it might well be. Researchers have developed a wide variety of mathematical and computer science tools which, when used in conjunction with other developments in physics and engineering, are related to a wide array of new digital technologies. These new technologies, such as bar codes, fax, email and voice mail, a wide array of new kinds of pagers, wireless telephony, ATM machines, digital television, the World Wide Web, etc., are changing the way people all over the world lead their lives and do business. Although they do not always require codes for their functioning, codes are in many ways directly related to these new technologies because codes can be used in a wide variety of information settings. They can be used to track, correct, compress, hide, and synchronize data, to name but a few of the more visible purposes that codes are put to. Diffie and Landau's book zooms in on a relatively narrow part of the information revolution but, none the less, a part of it that affects all people. Diffie and Landau zero in on privacy.

Concern with privacy is very ancient. From one perspective, to achieve privacy is to hide information that you do not want others to have. The need for keeping secrets in the affairs of state and the military are clear. Furthermore, there is a long tradition of using codes and other technical devices (e.g. secret writing) to achieve this secrecy. Julius Caesar is often credited with a great leap forward in the systematic attempt to hide information using codes. He devised the idea

[4]

that a plaintext could be disguised by replacing each letter of the alphabet with the letter of the alphabet obtained by shifting the the alphabet a fixed number of positions and cycling letters at the beginning around to the end (or vice versa). The goal even during early efforts in cryptography was the development of easy-to-use and impossible-to-break codes. (In technical parlance there is a difference between codes and ciphers, but here I will use the term codes in a generic fashion.)

From Caesar's early breakthrough, progress has accelerated. By the time World War II occurred most countries had sophisticated government agencies involved in the design and attempt to decipher codes. Going into the war the general public actually had the impression that by using ingenious mechanical machines, that military operations and government operations could be carried out without revealing information to the prying eyes of other countries. However, revelations made after the war exploded that myth. It was learned that British and American cryptographers had changed the course of the war and history by using mathematics and emerging computational techniques to break the German and Japanese codes. The significance of the fact that England was reading German codes was of such importance and value, that some have claimed that Churchill chose not to alert officials in Coventry of an impending German attack (which resulted in horrific loss of life and property) rather than risk that the Germans would deduce that their codes were compromised if Coventry displayed preparation for the well-guarded attack plans. (British government officials deny that Churchill did this, giving an explanation of why British authorities did not know of the Coventry attack plans. However, would authorities even today admit the true reasons if the decision involved trading the lives of innocent people for a "higher" good?)

These triumphs of human ingenuity raised the possibility that there was no such thing as an unbreakable cryptological system. This is not technically true. There is the one-time pad, a system which uses two copies of a randomly generated key to guarantee security. The problem with the one-time pad is that it requires a high overhead to implement. Thus, even though during the cold war the Soviet Union used one-time pads, the United States was able to take advantage of the sloppy implementation used.

Until a few years ago cryptology, the science of constructing and breaking codes, was largely within the realm of agencies of governments concerned with national security. In the United States this agency is the National Security Agency (NSA), an organization which until very recently worked in ways and at costs that were largely unscrutinized, certainly by the public at large and even by other parts of the government. While in recent years NSA was known for hiring large numbers of computer scientists, mathematicians, and language specialists, it did not have too much interface with other parts of national life. This was to change in part due to a remarkable theoretical breakthrough involving the management of code keys. First described in the open air of scholarly ideas, Whitfield Diffie, Martin Hellman and Ralph Merkle developed a revolutionary new approach to cryptography which involved what has come to be called public-key cryptography. Based on this innovation, other workers, notably Leonard Adleman, Ronald Rivest, and Adi Shamir, generated implementable systems based on these principles.

The development of public-key concepts not only raised the possibility of cheap secure codes but also other interesting possibilities. For example, when messages are sent there are concerns about issues such as who really sent the message or whether a message sent by X has been altered after X sent it. Can systems be devised that provide electronic equivalents of sharing power, signing a binding document, proving identity, etc.? Not all of the early suggestions for public-key systems have survived attempts to show that they could not be broken (i.e. Merkle and Hellman's knapsack system has been shown to be insecure). However, public-key ideas have raised the specter of easy to implement, secure (i.e. for all practical purposes unbreakable) codes that can be used to protect email, wireless telephony, and related technologies, including ones not yet thought of. Public-key

cryptography catapulted codes into the business world. In a global economy selling secure systems for the interchange of money or ideas between banks, credit card transitions, etc. offers a huge market. Suddenly, the activities of scholars and businesses came to the attention of NSA and/or the FBI. NSA was concerned that hardware of American origin would be sold abroad and used to protect the secrets of foreign countries whose secrets were currently potentially monitorable by the US. The FBI was concerned that members of organized crime would be able to avoid being brought to justice if they could take advantage of the security that new technologies might make possible.

This brings us to some details of Diffie and Laudau's book. The book begins with a brief history of cryptography and a primer of public-key methods and a discussion that new encryption ideas bring for a wide variety of improved and emerging technologies. (To support this discussion, I might have wished for an appendix that treated some of the details of the concepts behind public-key ideas.) The book then gives a detailed history of the way that law enforcement agencies have operated to obtain information that might be of value in solving or prosecuting crime. Specific attention is given to the issue of law enforcement agencies' being able to monitor and listen in on telephone traffic as compared with using surveillance devices and "wires" as sources of information. The authors detail the consequences that new encryption technology might have for law enforcement agencies and the public's desire for and perception of privacy. Might the general public prefer an assurance of communication privacy even when this means criminals would have the same protection as those who obey the law? Will key-escrow systems work as expected were society to decide that it wished to go this route? These kinds of questions and issues, as well as many related ones, are ably raised by this book.

Although Diffie and Landau are concerned about privacy issues, they do not bring to their readers' attention all aspects of new technology that affect this matter. Examples are the power of global positioning systems to monitor the location of people using cell phones, security via computer voice recognition systems, or finger print chips. The issues they raise, in fact, have broader settings than they suggest.

Since many feel that privacy is one of the core American values, Diffie and Laudau's book is a valuable service to all of us. It raises the important and subtle issues that scholars, legislators, and citizens have to balance in a capitalistic democracy. This well written and researched book deserves to be widely read.

References

Hardy, G.H., A Mathematician's Apology, Cambridge U. Press, New York, 1940.

Lebrow, I., The Digital Connection, Computer Science Press, Rockville, 1991.

Rivest, R., The Case against Regulating Encryption Technology, in Special Report, Computer Security and the Internet, Scientific American, October, 1998, p. 116-117.

Schneier, B, and D. Banisar, The Electronic Privacy Papers: Documents on the Battle for Privacy in the Age of Surveillance, Wiley, 1996.

Review of
A Theory of Objects
Series: Monographs in Computer Science
Authors: Martín Abadi and Luca Cardelli
Publisher: Springer-Verlag, 1996
ISBN: 0-387-94775-2
$44.95, Hardcover, 396 pages

Reviewer: Brian Postow

1 Overview

When a new programming language is designed it is sometimes useful to have a formal language in which we can describe with mathematical rigor what a program in that language is doing. This mathematical language can be thought of as a semantics of the programming language. If the semantics is sufficiently well developed we can use it to prove that a program does what we think it does. No actual programmer ever does this, but it is comforting to know that it can be done.

There are several different mathematical languages for giving the semantics of imperative programming languages (e.g. C or Pascal). Control flow graphs, or logical invariants can be used for this. There are also several different $\lambda-$calculi for describing the semantics of functional programming languages (e.g. Lisp or ML). Likewise we can use classical logic and resolution as a semantics of logical languages (e.g. Prolog). However, there is no mathematical language for discussing the semantics of object oriented programming languages (e.g. Java or Smalltalk) that is fully satisfying. The goal of this book is to fill this void, or at least to make a start.

The authors come from the functional programming and formal semantics side of the field, rather than the software engineering side, therefore they propose a calculus that is based on the $\lambda-$calculus, rather than any actual programming language in use. However, in this new calculus, which they call the $\varsigma-$calculus, instead of functions being primitive, objects are. The authors produce a family of $\varsigma-$calculi in order to hopefully represent more and more complicated object oriented techniques and structures.

2 Summary of Contents

The book is divided into 4 sections: a review of object oriented features, first order calculi, second order calculi, and higher order calculi.

2.1 Review: Object Oriented Features

This is a good overview of the object oriented structures and techniques that will be discussed in the rest of the book. The authors discuss the differences between class-based languages and object based languages, and the advantages and disadvantages thereof (the calculi in the book are almost all object based because of the increased simplicity). They also discuss subclassing, subtyping, inheritance, and subsumption, and explain what use and difficulties these will cause later in the book.

2.2 Part I: Untyped and First-Order Calculi

In Part I, the actual content of the book begins. The authors start by giving the simplest $\varsigma-$calculus, an untyped object calculus with no bells or whistles. Since this calculus doesn't describe subtyping at all, it is merely a jumping off point for the first order typed calculi.

Throughout the rest of the part, various issues are raised and constructs are added to the calculus to deal with them. For example, when subtyping is added to the calculus, it is found that function types don't follow the subtyping relation in the expected way. If you have 2 function types: $F = A \rightarrow B$ and $F' = A' \rightarrow B'$ and you want F to be a subtype of F' ($F < F'$), what do you need to know about A, A', B, and B'? Well, you want an object of type F to be usable in any place that expects an object of type F'. If $B < B'$ then whatever a function of type F returns, it will be acceptable to the receiver. This is called co-variance. On the other hand, when we look at the argument, the results are not as intuitive. Any argument that would be valid to pass to a function of type F' must be valid in the function of class F. Therefore $A' < A$ must hold. This is called contra-variance. Issues of co/contra-variance occur at regular intervals throughout the book.

2.3 Part II: Second-Order Calculi

Part I deals mainly with methods and sub-typing. Part II gets to the other main difficulty of object oriented semantics: the type of Self.

Many useful programs are very difficult to write if objects can't talk about their own type. However, determining the type correctness of objects that DO know what type they are becomes unintuitive very quickly. In addition, Self types add a variety of new variance issues. For example, what is the effect of inheritance on an inherited method that uses a Self type? This is a question that relates to much of the rest of the book.

2.4 Part III: Higher-Order Calculi

Having failed to come to a satisfying first or second-order calculi, the authors resort to higher-order calculi to deal with problems that they had developed earlier. They reference type theory, especially Girard's system F.

Using a higher order techniques, the authors do arrive at a calculus that allows them to do pretty much everything that they want to (including binary methods of the type Self * Self $\rightarrow$ Self). However at this point the calculus is so complex and cluttered with syntax that it becomes very difficult to work with.

3 Style

This is not an easy book. The overview in Part 0 is relatively easy, and should be readable to anyone who knows a little bit about what object oriented programming is. Part I is a little more challenging and assumes some familiarity with the $\lambda-$calculus, and the concept of a type derivation. Part II is more difficult with more reliance on type theory. Part III gets very dense, with pages of complex type derivations. However, to make it easier the same 3-5 examples are used repeatedly throughout the book, so it is easy to compare calculi.

4 Opinion

This book doesn't claim to be a complete theory of objects, merely a first attempt. At this it succeeds. The theory that the authors develop is much more complex than would be hoped. For example, the list of notations used in the book takes 15 pages, and there are over 30 different calculi and languages described in the book.

Over all, it is a very interesting book, but the authors don't quite live up to their goal of making a simple language for discussing object oriented issues.