

The Book Review Column¹
by William Gasarch
Department of Computer Science
University of Maryland at College Park
College Park, MD, 20742
email: gasarch@cs.umd.edu

In this column we review the following books.

1. **Bioinformatics: The Machine Learning Approach** by Pierre Baldi and Soren Brunak. Reviewed by Ivelin Ivanov. This book on Bioinformatics is aimed at both biologists and computer scientists and is an attempt to build a bridge between the two.
2. **The Clausal Theory of Types** by D. A. Wolfram². Reviewed by Maulik Dave. The book describes the Clausal Theory of Types (CTT). CTT is related to Church's Simple Theory of Types which is a theory of types on the lambda Calculus. Its role as a logic programming language incorporates functional programming.
3. **Discrete Mathematical Problems with Medical Applications (DIMACS Volume 55)** Edited by D-Z. Du, P. M. Pardalos, and J. Wang. Reviewed by Suresh Venkatasubramanian. This is a collection of articles from a DIMACS workshop about using discrete math for medical applications. Applications include Medical Imaging, Classification, Mining, and Radiosurgery.
4. **Things a Computer Scientist Rarely Talks About** by Donald E. Knuth. Reviewed by Ian Parberry. This is a summary of six lectures Don Knuth gave on interplays between Computer Science and Religion.
5. **A New Kind of Science** by Stephen Wolfram. Review by Hassan Masum. Its not new, its not science, and its not kind.

Books I want Reviewed

If you want a FREE copy of one of these books in exchange for a review, then email me at gasarch@cs.umd.edu

Reviews need to be in LaTeX, LaTeX2e, or Plaintext.

Books on Algorithms, Combinatorics, and Related Fields

1. *Diophantine Equations and Power Integral Bases* by Gaal.
2. *Algorithm Design: Foundations, Analysis, and Internet Examples* by Goodrich and Tamassia.
3. *An Introduction to Data Structures and Algorithms* by Storer.
4. *Computational Line Geometry* by Pottmann and Wallner.
5. *Linear Optimization and Extensions: Problems and Solutions* by Alevras and Padberg.

Books on Cryptography and Books on Learning

¹© William Gasarch, 2002.

²Not to be confused with Stephen Wolfram, whose book is also reviewed in this column.

1. *Elliptic Curves in Cryptography* by Blake, Seroussi, and Smart.
2. *Coding Theory and Cryptograph: The Essentials* by Hankerson, Hoffman, Lenoard, Linder, Phelps, Rodger, and Wall.
3. *Learning with Kernels (Support Vector Machines, Regularization, Optimization, and Beyond)* Bernard Scholkopf and Alexander Smola.
4. *Learning Kernel Classifiers* by Herbrich.

Books on Complexity and Logic

1. *Models of Computation: Exploring the Power of Computing* by John Savage.
2. *Complexity and Information* by Traub and Werschulz.
3. *Derivation and Computation* by Simmons.
4. *Logic and Language Models for Computer Science* by Hamburger and Richards.
5. *Automata Theory and its Applications* by Khoussainov and Nerode.
6. *Introduction to Language, Machines, and Logic* by Parkes

Review of: **Bioinformatics: The Machine Learning Approach** ³

by Authors: Pierre Baldi and Soren Brunak

Publisher: MIT Press

\$50.00, 400 pages

1 Overview

Undoubtfully, bioinformatics is one of the fastest developing scientific disciplines in recent years. Bioinformatics is the development and application of computer methods for management, analysis, interpretation, and prediction, as well as for the design of experiments.

There is already a significant number of books on bioinformatics. Some are introductory and require almost no prior experience in biology or computer science: "Bioinformatics Basics Applications in Biological Science and Medicine" and "Introduction to Bioinformatics". Others are targeted to biologists entering the field of bioinformatics: "Developing Bioinformatics Computer Skills". Some more specialized books are: "An Introduction to Support Vector Machines : And Other Kernel-Based Learning Methods", "Biological Sequence Analysis : Probabilistic Models of Proteins and Nucleic Acids", "Pattern Discovery in Bimolecular Data : Tools, Techniques, and Applications", "Computational Molecular Biology: An Algorithmic Approach".

The book subject of this review has a broad scope. "Bioinformatics: The machine learning approach" is aimed at two types of researchers and students. First are the biologists and biochemists who need to understand new data-driven algorithms, such as neural networks and hidden Markov

³© Ivelin Ivanov, 2002.

models, in the context of biological sequences and their molecular structure and function. Second are those with a primary background in physics, mathematics, statistics, or computer science who need to know more about specific applications in molecular biology.

I must agree with other reviews (<http://www.dlib.org/dlib/september98/09bookreview-2.html> and one of the reviews on amazon) on this book noting that the book has a problem of properly selecting the audience. Since bioinformatics concerns the application of advanced computer science and probability theory to the most recent discoveries in molecular biology, the question arises, what basic knowledge should the reader have? The technical prerequisites mentioned are "basic calculus, algebra, and discrete probability theory" at an undergraduate level, while any prior knowledge in the molecular biology domain is not required. The cover even claims that the book can be used as an introduction into bioinformatics, by both researchers and students with only a primary background in either one of those areas. At least some knowledge of molecular biology is required for a proper understanding of the (sometimes rather dense) material presented, and that some experience in applying probability theory would be very helpful as well.

2 Content Summary

Chapter 1- Introduction

Provides introduction to sequence data in the context of molecular biology, and sequence analysis. It contains in particular an overview of genomes and proteomes, the DNA and protein "universes" created by evolution that are becoming available in the public databases. It presents an overview of genomes and their sizes, and other comparative material that, if not original, is hard to find in other textbooks.

Chapter 2- Machine Learning Foundations: The Probabilistic Framework

A key theoretical chapter. Lays the foundations for all machine-learning techniques, and shows explicitly how one must reason in the presence of uncertainty. It describes a general way of thinking about sequence problems: the Bayesian statistical framework for inference and induction. The main conclusion derived from this framework is that the proper language of machine learning, and for addressing all modeling problems, is the language of probability theory.

Chapter 3- Probabilistic Modeling and Inference: Examples

Demonstrates the Bayesian probabilistic framework with a set of classical examples which are used in following chapters.

Chapter 4- Machine Learning Algorithms

Discusses many of the basic algorithms required for Bayesian inference, machine learning, and sequence applications, in order to compute expectations and optimize cost functions.

Chapter 5- Neural Networks: The Theory

Chapters 5-9 and Chapter 12 form the core of the book. Chapter 5 provides an introduction to the theory of neural networks. It contains definitions of the basic concepts, a short derivation of the "backpropagation" learning algorithm, as well as a simple proof of the fact that neural networks, which are often introduced without any reference to probability theory, are in fact best viewed within the general probabilistic framework of Chapter 2. This in turn yields useful insights on the design of neural architectures and the choice of cost functions for learning.

Chapter 6- Neural Networks: Applications

Contains a selected list of applications of neural network techniques to sequence analysis problems. The authors have selected seminal where advances in the methodology have provided significant improvements over other approaches. A special treat is given to the issue of optimizing training procedures in the sequence context, and how to combine networks to form more complex

and powerful algorithms. The applications treated in detail include protein secondary structure, signal peptides, intron splice sites, and gene-finding.

Chapter 7- Hidden Markov Models: The Theory

Chapters 7 and 8, on hidden Markov models, mirror Chapters 5 and 6. Chapter 7 contains a fairly detailed introduction to hidden Markov models (HMMs), and the corresponding dynamic programming algorithms (forward, backward, and Viterbi algorithms) as well as learning algorithms (EM, gradient-descent, etc.). Hidden Markov models of biological sequences can be viewed as generalized dice models with insertions and deletions.

Chapter 8- Hidden Markov Models: Applications

Contains a selected list of applications of hidden Markov models to both protein and DNA/RNA problems. It demonstrates, first, how HMMs can be used, among other things, to model protein families, derive large multiple alignments, classify sequences, and search large databases of complete or fragment sequences. In the case of DNA, authors show how HMMs can be used in gene-finding (promoters, exons, introns) and gene-parsing tasks.

Chapter 9- Probabilistic Graphical Models in Bioinformatics

HMMs can be very effective, but they have their limitations. Chapters 9-11 can be viewed as extensions of HMMs in different directions. Chapter 9 uses the theory of probabilistic graphical models systematically both as a unifying concept and to derive new classes of models, such as hybrid models that combine HMMs with artificial neural networks, or bi-directional Markov models that exploit the spatial rather than temporal nature of biological sequences. The chapter includes applications of gene-finding, analysis of DNA symmetries, and prediction of protein secondary structure.

Chapter 10- Probabilistic Models of Evolution: Phylogenetic Trees

Presents phylogenetic trees and, consistent with the framework of Chapter 2, the inevitable underlying probabilistic models of evolution.

Chapter 11- Stochastic Grammars and Linguistics

Covers formal grammars and the Chomsky hierarchy. Stochastic grammars provide a new class of models for biological sequences, which generalize both HMMs and the simple dice model. Stochastic regular grammars are in fact equivalent to HMMs. Stochastic context-free grammars are more powerful and roughly correspond to dice that can produce pairs of letters rather than single letters. Applications of stochastic grammars, especially to RNA modeling, are briefly reviewed.

Chapter 12- Microarrays and Gene Expression

Focuses primarily on the analysis of DNA microarray gene expression data, once again by generalizing the die model. Shows how the Bayesian probabilistic framework can be applied systematically to array data. In particular, we treat the problems of establishing whether a gene behaves differently in a treatment versus control situation and of gene clustering. Analysis of regulatory regions and inference of gene regulatory networks are discussed briefly.

Chapter 13- Internet Resources and Public Databases

Contains an overview of current database resources and other information that is publicly available over the Internet, together with a list of useful directions to interesting WWW sites and pointers. Because these resources are changing rapidly, we focus on general sites where information is likely to be updated regularly. However, the chapter contains also a pointer to a page that contains regularly-updated links to all the other sites.

Appendix A- Statistics

The book contains in appendix form a few technical sections that are important for reference and for a thorough understanding of the material. Appendix A covers statistical notions such as errors bars, sufficient statistics, and the exponential family of distributions.

Appendix B- Information Theory, Entropy, and Relative Entropy

Focuses on information theory and the fundamental notions of entropy, mutual information, and relative entropy.

Appendix C- Probabilistic Graphical Models

Provides a brief overview of graphical models, independence, and Markov properties, in both the undirected case (random Markov fields) and the directed case (Bayesian networks)

Appendix D- HMM Technicalities, Scaling, Periodic Architectures, State Functions, and Dirichlet Mixtures

Covers technical issues related to hidden Markov models, such as scaling, loop architectures, and bendability.

Appendix E- Gaussian Processes, Kernel Methods, and Support Vector Machines

Briefly reviews two related classes of machine learning models of growing importance, Gaussian processes and support vector machines.

3 Conclusion

"Bioinformatics - The Machine Learning Approach" is a valuable reference work and update on developments in bioinformatics. It is, however, less appropriate as a general introduction into the area of bioinformatics. The book may be even more valuable if the authors consider to relax the conciseness and replace some of the references to external resources throughout the book with intuitive inline explanations of the referenced material.

Review of
The Clausal Theory of Types
Cambridge University Press 1993
by **D. A. Wolfram**

Published in 1993 by the Press Syndicate of the University of Cambridge, 124 pages

Reviewer: Maulik Dave⁴

1 Overview

The book describes the Clausal Theory of Types (CTT). The CTT is a clausal extensional sub-logic of Church's formulation of the Simple Theory of Types. It is based on a lamda-calculus formulation of a clausal logic with equality. Its role as a logic programming language incorporates functional programming.

2 Content Summary

The book has five chapters.

Chapter 1 presents logic programming as a case study. It starts with the Skolem-Herbrand-Godel theorem of first-order predicate calculus. The chapter traces the developments which led to the logic programming from automations of theorem proving. The chapter presents an approach, which is a good introduction for the rest of the book.

⁴© Maulik Dave, 2002

Chapter 2 is on simply typed lambda calculus. The simply typed lambda calculus is the syntactic basis of Church's formulation of higher-order logic. The chapter supplies necessary information on type symbols, terms, conversions and reductions, normal forms, and substitutions.

After building background in chapter 1, and 2; the chapter 3 introduces Clausal Theory of Types. The outcome of the chapter 3 is higher order Skolem-Herbrand-Godel theorem for the CTT.

Chapter 4 presents details related to the higher-order equational unification. Major forms and their complexities down to the first-order cases are presented. Higher order equational unifiability uses a definition of higher-order rewriting. Soundness and completeness results for higher-order equational unification procedures are given. Higher-order resolution with built-in higher-order equational theories for the Clausal Theory of Types formulas is defined. Also, higher-order unification and matching, second-order monadic unification, and first-order equational unification are discussed.

Chapter 5 is on higher-order equational logic programming. The chapter discusses the role of CTT as a programming language. It provides a method for testing the unsatisfiability of CTT formulas. It defines CTT horn clauses, and shows that they meet the programming language criteria of being computationally adequate, and possessing least models. The chapter goes on to define operational semantics of CTT programs. It concludes with discussion of a prototype interpreter for CTT programs.

3 Conclusion and Comments

The book is highly theoretical in nature of descriptions. There are occasional examples found to explain the theories. The marriage between the automation of theorem proving, and the logic programming has come out of the book strongly. The book is useful for interests like theory of logic programming, semantics of the logic programming languages, automation of theorem proving. With some efforts, the book can also be used for practical purposes related to the programming language - CTT.

Discrete Mathematical Problems with Medical Applications
DIMACS Volume 55

Edited by D-Z. Du, P. M. Pardalos, J. Wang

Published by American Mathematical Society

ISBN: 0-8218-2096-6

219 pages

Review by Suresh Venkatasubramanian⁵

AT&T Labs – Research

1 Introduction

The influence of computer science and discrete mathematics has long been felt in the world of the life sciences. Fields like computational biology and chemistry have been a rich source of problems in string matching, graph theory, algorithms, and computational geometry, and there have been extensive interactions between researchers in the life sciences and computer scientists in these areas.

Medical informatics is another area that has had very fruitful interactions with computer science. Some of the first expert systems were developed as diagnostic aids (to help diagnose various kinds

⁵© Suresh Venkatasubramanian 2002

of medical conditions), and today, with the existence of sophisticated computer hardware to assist in diagnostic and surgical procedures, there is even more of a need for tools to help analyse data and plan procedures.

This book is a collection of papers from the DIMACS Workshop on Discrete Mathematical Problems with Medical applications, and provides a glimpse of some of the problems that lie across the medicine/computer science divide. The underlying focus of the papers is *digital data*: to start backwards, the last two papers in the collection (by J. K. Udupa and P. Hall) are position papers discussing the premise that ‘...scenes are inherently digital, [and so] we should take entirely digital approaches to realizing all 3D imaging operations’.

2 Medical Imaging

A bulk of the papers in the collection address issues in the domain of medical imaging. Typical sources of digital images in this domain are X-ray photographs, tomographic slices, and magnetic resonance images. There are also sources such as time-series data from heart monitors and electroencephalographic (EEG) recordings. The basic problem of medical imaging can be loosely described as follows:

Given a set of images representing views of internal body structure or activity, reconstruct the shape (if any) of the structure being observed, and use this shape to aid in diagnosis (for example by detecting anomalous structures).

Variants of this process occur when the data being gathered does not represent internal structure but measures internal activity (EEGs are one such example). In such cases, the reconstruction step is not relevant, but the problem of modelling the signal being captured and identifying interesting patterns is important (for example to identify the onset of epileptic seizure).

The paper by Nyström and Smedby (pg. 75) provides an example of the use of shape representation in magnetic resonance imaging. They illustrate how the *medial axis* can be used to represent capillary structure as detected by magnetic resonance angiograms, and can be used to diagnose irregularities in the vascular system. The following paper by Bouix and Siddiqui (pg 91) presents a general method for medial axis computation in the context of such applications, and notes that “despite their popularity, the stable numerical computation of medial surfaces remains a challenge”.

A related paper by Hall (pg 11) addresses the issue of representation in the vascular system, and provides a model that combines the combinatorial structure of the arteries and veins with geometric information about them, as well as a measure of similarity. One could imagine techniques like medial surface computations being used to *acquire* the models that this paper describes.

Other forms of representation, progressively less complex, are described in the paper by Rangarajan and Chui (pg 105) and Kuba *et al.* (pg 195). In the former, image registration methods (borrowed from computer vision) are used to compute similarity between MR images, and in the latter paper, the problem of acquiring shape is viewed as a general tomographic problem i.e determine (approximately or partially) an unknown function defined over the reals from weighted sums over subsets of its domain.

2.1 Classification and Mining

Other papers in this volume discuss classification and mining procedures to identify anomalous structures or signals within the body. The paper by Lee *et al.* (pg 1) uses a support vector approach

to determine viable candidates for chemotherapy (for breast cancer) from information revealed by removing portions of malignant tumors from patients. Sophisticated dynamical systems theory is used by Iasemidis *et al.* (pg 55) to aid in the prediction of epileptic seizures from EEG activity.

3 Radiosurgery

In radiosurgery, concentrated bursts of radiation are directed at specific targets (like tumors) to irradiate them. The success of such a process depends on the ability of the radiation device to provide highly focused doses of radiation in specific areas in such a way as to maintain a minimum and maximum level of dosage in the tumor and surrounding tissue. Systems like the Gamma Knife and Multi-Leaf Collimators provide such functionality, and the desired dosage distribution is achieved by superposing dosage from multiple target sites (centers). The challenge therefore is to determine a set of centers (given the radiation beam characteristics) such that the resulting radiation levels mirror the desired distribution, both in terms of providing a minimum level of radiation to the targeted areas and ensuring a (much smaller) maximum level of radiation to surrounding tissue areas.

The papers by Ferris and Shephard (pg. 27), Wu (pg. 45), and Wu *et al.* (pg. 183) address this optimization problem. Variations in the approaches taken come from the function to be optimized (number of centers, total root-mean-square error between distributions), model of the beam (a continuous distribution or a uniform disc of bounded radius), and the radiation device itself (the Gamma Knife or the Multi-Leaf Collimator).

4 Remarks

One of the central themes of this collection is the application of imaging methods in the domain of medicine. More generally, there is a focus on the manipulation of *digital data* and the extraction of higher-level semantic information from it, much the same activity that informs fields like computer vision and image recognition. This area provides a challenging and interesting set of tasks for computer scientists (especially those working in computational geometry, applied geometry and graphics) and the potential for very fruitful interdisciplinary efforts is very high. Computational geometers will find the main methods used in this book (such as the use of the medial axis of a shape) quite familiar; David Eppstein's page describing the various settings in which the medial axis of a polygon has been used (<http://www.ics.uci.edu/~eppstein/gina/medial.html>) is also very helpful. The papers themselves explain in considerable detail the underlying medical domain knowledge that a computer scientist might need to understand in order to grasp the problems being studied.

Although the main focus is on digital data, there are also other papers in the collection that deal with more traditional topics from computational biology, such as combinatorial group testing as applied to DNA Library screening, and computing the nearest neighbour interchange distance.

Review of
Parallel Processing and Parallel Algorithms: Theory and Computation ⁶
by **Seyed H. Roosta**
Springer Verlag, 650 pages, \$62.50
Review by **Ivelin Ivanov** ivelin@acm.org

1 Overview

This book covers the essential elements of parallel processing and parallel algorithms. It is unique in that it is a self-contained book. It covers everything fundamental to parallel processing from computer architecture to parallel programming and parallel algorithms. It is designed to function as a text for an undergraduate course in parallel processing, but also works well as a comprehensive reference for professionals interested in all phases of parallel processing and parallel programming.

2 Content Summary

This book aims to cover one aspect of the analysis of parallel computers, which is the essence of architectures. It attempts to cover the relationship between parallel programming approaches and machines, or algorithms and architectures.

Chapter 1 (Computer Architecture) is a broad overview of the different classes of parallel hardware architectures. It emphasizes the fact that understanding and selecting parallel programming concepts depends on the underlying computer architecture they're applied against. In simple and clear words it explains some basic acronyms often used in today's computer literature: SISD (Single Instruction Stream, Single Data Stream), SIMD (Single Instruction Stream, Multiple Data Stream), MISD (Multiple Instruction Stream, Single Data Stream), MIMD (Multiple Instruction Stream, Multiple Data Stream).

Chapter 2 (Components of Parallel Computers) introduces basic parallel computer terminology including memory, interconnection networks, compilers, operating systems and input/output constraints.

Chapter 3 (Principles of Parallel Programming) examines the principles of parallel programming with regard to mapping the algorithms to specific architectures. It compares message passing with shared-address-space parallelism and studies levels of parallelism and granularity problems.

Chapter 4 (Parallel Programming Approaches) presents known programming languages for transforming sequential programs into parallel forms. The parallel programming approaches are applied to various parallel architectures.

Chapter 5 (Principles of Parallel Algorithm Design) is an introduction to parallel algorithm design, performance measures and complexity. It presents five basic principles in the design of parallel algorithms:

1. The Brent Scheduling Principle, which allows reducing the number of processors in existing parallel algorithms, without increasing the total execution time.
2. The Pipelining Principle, which can be used when certain steps in the execution of an algorithm can overlap.
3. The Divide and Conquer Principle of splitting a problem into several small independent components and solving them in parallel.

⁶© 2002 Ivelin Ivanov

4. The Dependency Graph Principle of creating a directed graph, where nodes represent blocks of individual operations and the edges represent situations in which one block of operations depends on the outcome of performing other blocks.
5. The Race Condition Principle stating that if two processes try to access the same shared data, they may interfere with each other.

The authors dedicated the next three chapters to a variety of parallel algorithms. They have selected problems with inherently parallel nature and have efficient parallel solutions.

Chapter 6 (Parallel Graph Algorithms) treats a variety of parallel graph algorithms. The chapter is devoted to a discussion of a number of basic graph problems and presents the design and analysis of efficient parallel algorithms for them. It covers algorithms for identifying connected components in a graph, paths and all-pairs shortest paths, minimum spanning trees and forests, the traveling salesman problem, detecting cycles in a graph and coloring of graphs.

Chapter 7 (Parallel Search Algorithms) focuses on the most prominent search methods in the context of parallel processing. It deals with two basic problems concerned with finite lists of elements: selection and searching.

that are of fundamental importance in scientific computations. It classifies parallelism in computational algorithms that demonstrate various types of parallelism such as sorting, matrix multiplication and systems of linear equations.

Chapter 9 (Data Flow and Functional Programming) describes data flow computing, which is a fundamentally different issue in parallel processing and can be achieved by data flow languages. The purpose of this contribution is to deal with the architecture of data flow computers. SISAL is a functional language without explicit parallel control constructs, but its data types and its constructs for expressing parallelism and operations are specifically chosen to produce efficient code for large-scale scientific computations.

Chapter 10 (Asynchronous Parallel Programming) Studies asynchronous computations when all program fragments are initially regarded as parallel, independent and unordered. Any constraint on their interactions is formulated as explicit or implicit individual conditions associated with fragments. The chapter introduces three different languages, of which Modula-2 is used in hybrid systems of SIMD-MIMD computers.

Chapter 11 (Data Parallel Programming) introduces a high-level notation that simplifies parallel programming and enhances portability. This provides compilers with enough information to allow them to generate efficient parallel code for both shared-memory multiprocessors and distributed-memory multi-computers. The languages presented are Dataparallel C, a variant of the original C* language developed by Thinking Machines Corporation and Fortran 90, which has vector and array operations.

Chapter 12 (Artificial Intelligence and Parallel Processing) discusses knowledge processing, which is fast-growing area of computer-science and engineering and how utilizing parallel processing may increase the processing speed of rule-based expert systems. It discusses Concurrent Prolog as a parallel logic language and Multilisp, which is a modification of an existing language (Lisp). The basic idea behind Multilisp is parallel expression evaluation.

3 Conclusion

This book progresses from theory to computation, exploring the fundamentals of parallelism and the relationship between parallel programming approaches, algorithms, and architectures.

I wouldn't think it could be one's first reading in the field of parallel computation. However, with some real world experience in sequential software design and inspiration for applying parallelism, this book can be a very rich and educational resource. It is a bright light tower in the adventurous world of distributed and parallel computing.

Finally, I thought it is worth mentioning that the book received the highest rating from readers who published their reviews on the web site of a very popular online book retailer.

Review of

DONALD E. KNUTH, *Things a Computer Scientist Rarely Talks About*

The University of Chicago Press, 2001, 257 pages.

Reviewed by Ian Parberry⁷, University of North Texas

I was initially excited to receive this book to review, given that I've occasionally had thoughts and even published an obscure paper about how complexity theory can tell us something about real life (in my case, knowledge and understanding). Knuth's book consists of transcripts of a series of six lectures given by Knuth at MIT in 1999 followed by the transcript of a panel discussion on Creativity, Spirituality, and Computer Science by Harry Lewis, Guy Steele, Manuela Veloso, Donald Knuth, and Mitch Kapur.

I was a little disappointed that Knuth spent a large amount of his lectures talking about his book *3:16*. The basic idea behind *3:16* was that Knuth used ideas from random sampling and zero-knowledge proofs to study the Bible. Instead of studying all of it, or even a truly random sample of it, he chose for reasons that are probably understandable to most complexity theorists (but probably not to Biblical scholars) to study a deterministically chosen stratified set of pseudorandom verses: Chapter 3, verse 16 of each book of the Bible.

Each of the six lectures ends with a question-and-answer session followed by bibliographic notes. Lecture 1 was his Introduction. Lecture 2 described the pros and cons of his random sampling technique. Lecture 3 described the pros and cons of doing his own translations from Greek and Hebrew. Lecture 4 mostly talked about calligraphy. Lecture 5 compared and contrasted what Knuth learned about God and what Knuth learned about theology from working on *3:16*. Lecture 6 was the meat of the book for me, discussing how concepts of computer science including computational complexity might give insights about divinity. The transcripts of the lectures are followed by the transcript of the panel discussion, some bibliographic notes on the panel discussion, an index of Biblical references, and a full and copious 15-page index of the complete volume (an excellent Index in the style we have come to be accustomed to from a master such as Knuth.)

I am a little biased in my review because I disagree with the basic premise of publishing a book that is a transcription of a series of lectures. While it is a way to give a secondhand experience to people who were unable to attend, it of course doesn't give the full experience of gesture and nuance, and some of the verbal flourishes that are useful in a lecture are unnecessary and even distracting in book form. I think I would prefer to see a DVD of Knuth delivering the lectures. I particularly disliked the transcripts of the question-and-answer sessions for the admittedly selfish reason that I didn't like most of the questions and I would have asked different ones.

Religion is one of the things that computer scientists rarely talk about, possibly because we are prevented from raising such issues in class due to separation of church and state in the US. *Should* computer scientists talk about these things? I believe that we should all avoid the Transfer

⁷© Ian Parberry, 2002

of Authority phenomenon, in which members of the public assume that just because a scientist is an authority on one thing (such as theoretical computer science) makes him or her an authority on another totally unrelated issue (such as religion). I am happy to report that Donald Knuth is both an expert theoretical computer scientist and has some cogent things to say about religion.

Review of *A New Kind of Science*⁸

Stephen Wolfram

Publisher: Wolfram Media, 2002

ISBN: 1579550088

Review by:

Hassan Masum

Carleton University, Ottawa, Canada

hmasum.com

1 In a Nutshell

A New Kind of Science uses a wide range of easy-to-understand models - mostly cellular automata variants - to explore one key idea: some simple computational models can generate "very complex" behavior, and may describe aspects of the physical and mathematical universe better than traditional mathematical approaches.

Wolfram's book has been the subject of much speculation, partly due to its lengthy preparation time (10 years). Since it came out a few months ago, published opinions of the book have ranged from "worthless" to "ground-breaking".

As you will glean from points made later, this book is a difficult one to review objectively, for two main reasons: i) the writing style, and ii) the sheer breadth of questions raised. What follows is therefore a personal description of how the book did and did not impress me.

On the positive side, Wolfram raises many interesting questions. He specifies a remarkably diverse range of models related to cellular automata variants. And he explains his ideas simply, with beautiful pictures and a main text which is accessible to any reader with a high-school education who is willing to think a bit.

On the negative side, many of the ideas in the book have been discussed elsewhere, so there is not much material that is fundamentally new (or immediately usable). Some of the reasoning (such as the lack of differentiation between degrees of complexity) seems incorrect. And the immodest writing style and relative lack of acknowledgment and credit to other researchers is distasteful.

While it asks many interesting questions in a clear if sometimes overstated way, this book does not create "a new kind of science". It does, however, bring together many key issues and simple models for the interested reader - and may well stimulate research into the questions and models proposed.

2 So What's it about Anyway?

The main text consists of 12 chapters taking up the first 3/4 of the book. (This main text is in a rather sparse typesetting layout - the book could probably have been 300 pages shorter with less

⁸© Hassan Masum, 2002

line spacing and margins). Most chapters cover a wide breadth of ideas - the following is a fairly lossy compression:

1) The Foundations for a New Kind of Science. An advertisement for the rest of the book. Skip directly to chapter 2.

2) The Crucial Experiment. Looks at the evolution of 1-dimensional binary cellular automata. In the simplest case where the state of a cell at time $t+1$ depends only on the state of itself and its two neighbors at time t , there are 256 possible rules; while many yield repetitive or nested behavior, a few yield behavior that seems random or complex in a non-obvious way. (Interesting in its own right, but Wolfram claims it to be a conceptual revolution that simple programs can generate complex behavior.)

3) The World of Simple Programs. 1D cellular automata, mobile automata, Turing machines, substitution systems, tag systems, register machines - with many clear diagrams, it's shown that "complex behavior" can arise with simple initial conditions in all these systems. OK, that's neat - looking at the diagrams there is output with no obvious patterns. But then it is mentioned that past this low threshold, adding complexity to the rules doesn't lead to a perceptible increase in complexity - and thereafter this observation is used operationally to equate "complex" with "seeming complex via visual perception and passing statistical tests for randomness" (discussed further later in the book). An interesting note is made of how the discoveries were made; two principles are i) "the best computer experiments are ones that are as simple and straightforward as possible", and ii) good visual displays of data greatly aid pattern-finding.

4) Systems Based on Numbers. Not just cellular automata type systems, but numeric systems as well can yield seemingly complex behavior. Examples: looking at powers of 3 in base 2 yields a seemingly random pattern of digits, variations of the $3n+1$ problem, recursive sequences like $f(n)=f(n-f(n-1))+f(n-f(n-2))$, mathematical constants, continued fraction representations, iterated maps, continuous cellular automata and partial differential equations. The point is that "complex behavior" exists even in numeric and continuous systems.

5) Two Dimensions and Beyond. Generalizing beyond one dimension: two and three dimensional cellular automata, fractals, network systems with general inter-element connectivity, multiway systems (i.e. nondeterministic computation). Constraint-based systems - differentiating between systems generated by explicit construction rules vs implicitly to satisfy given constraints.

6) Starting from Randomness. Spontaneous order from random initial conditions. Wolfram talks about his four-class system classification: 1) static final state, 2) cyclic final state, 3) random behavior, 4) "complex" behavior. Sensitivity to initial conditions, basins of attraction.

7, 8) Mechanisms in Programs and Nature, Implications for Everyday Systems. Discussion of applications to natural systems. Randomness, discrete vs continuous systems, satisfying constraints. Snowflakes, fluid flow, form and pattern in biology, natural selection.

9) Fundamental Physics. Some intriguing ideas about physics as computation - it's the "universe as a cellular automaton" idea in detail. It's anybody's guess whether or not these ideas will eventually bear fruit.

10) Processes of Perception and Analysis. Perception, statistics, mathematics. There is discussion in this chapter of the issue of deciding that a pattern is "complex" by visual pattern recognition, as opposed to by statistical tests, computational analysis, or some other methods. While the issue is addressed to some extent with acknowledgment of specific weaknesses of human visual pattern recognition, I believe it needs more exploration (and before the tenth chapter in the book). The core of many of the previous chapters in the book is saying "look, here is a simple

system which makes a 'remarkably complex' pattern", which is a weak argument if the definition of 'remarkably complex' varies substantially across observers. (If we are going by visual inspection, I find e.g. some fractals and other such generative patterns to be more visually complex than most of the patterns in the book - and let's not even get into agent-based or artificial life simulations.)

11) The Notion of Computation. Universal computation. Very simple examples of cellular automata and several other systems can simulate a Universal Turing Machine. This is nice, but not necessarily too relevant in the greater scheme of things - Wolfram does recognize that higher-order information like (speaking loosely) the expected probability that a random pattern will do some kind of complex computation may be more important for many purposes than just the existence of some arcane TM simulation, but again this issue feels underexplored.

12) The Principle of Computational Equivalence. This final chapter is devoted to a lengthy discussion of Wolfram's proposed "Principle of Computational Equivalence", which Wolfram states in a general form as "almost all processes that are not obviously simple can be viewed as computations of equivalent sophistication". The first step toward this principle is noting the empirical existence of computation universality (Church-Turing thesis). Next, he goes further and suggests that, no matter how simple or complicated either the rules or initial conditions for a process are, it will almost always correspond to a computation of "equivalent sophistication" as long as the observed process "does not look obviously simple". It is not really possible to summarize the 100+ pages in which this thesis is argued - to me it feels something like the idea of computational depth writ large. Maybe it is true, but it will take a lot of work to establish it. The section "Implications for Mathematics and Its Foundations", pp 772-822, may be of particular interest (although some of it is presaged by Chaitin and others).

Notes. After completing the text proper (227580 words), one is less than halfway through - for the last 1/4 of the book has more words (283751) than the entire body proper, in notes to the main text written in a smallish type size. (It seems to be only in these notes that anyone other than Wolfram is acknowledged.) The notes do provide a lot of specific avenues of exploration to follow up on, and many relevant - and usually quite well-written - explanations of related concepts.

Personally, I really would have liked some of this material integrated with the main text - it would have made the exposition feel more solid, and integrated it better with the rest of humanity's intellectual output. In fact, I often found the notes to be better reading than the text proper. They contain a huge variety of topics covered in a competent and clear way, and are quite enjoyable to browse through.

I'll close this section with a quote from the notes, on a principle that Wolfram has clearly lived up to: "Probably the single most important principle that I have followed is just to try to keep everything as simple as possible. Study the simplest systems. Ask the most obvious questions. Search for the most straightforward explanations. For among other things, this is ultimately how the most useful and powerful results are obtained. Not that it is easy to do this. For while in the end it may be possible to get to something simple and elegant, it often takes huge intellectual effort to see just how this can be done. And without great tenacity there is a tremendous tendency to stop before one has gone far enough."

3 The Good

Print quality is excellent, and the pictures are beautiful and often thought-provoking. The clarity of the writing is remarkably high, given the subject material. A great deal of effort clearly went into making the book visually appealing and easy to understand, for which much credit is due.

A remarkable variety of models are introduced, specified, graphically illustrated, and discussed. Most of these models are variants or generalizations of the usual cellular automata (CA) models - while Wolfram does spend a lot of time talking about 1-dimensional CAs, he also discusses 2D and 3D CAs, non-hypercube connectivity, continuous CAs, "mobile automata", many-state CAs, and so on. Although the models ultimately seem too simplistic to do more than suggest the kind of complex behavior that Wolfram claims they possess, they do form a jumping-off point for anyone interested in playing around with this class of computational structures.

Wolfram asks many important questions. (Many have been asked before, but many may not have been, and it is nice to see them all in one place and explained clearly.)

Many handwaving books about "complexity" or "complex systems" for a popular audience focus more on personalities and stories than science. Here, the focus is mostly on a set of models which are explained concisely and illustrated profusely. Most of the models are easily replicable (many have Mathematica code snippets to describe the basic logic).

One has to appreciate Wolfram's use of simple systems, and the graphical display methods used (which are sometimes quite clever in showing particular phenomena, in a way that reminded me of Edward Tufte's visual illustrations). The sheer amount of work that must have gone into generating and playing around with all the models is worth noting.

This book is a significant attempt to integrate theory, observations, and speculations on the computational behavior of simple models. It is, however, flawed...

4 The Not-So-Good

Right from the start, one is subject to many repetitions of phrases such as "the new kind of science in this book", "the discoveries in this book", and other similar phrases. Wolfram claims to have discovered what was already done elsewhere, e.g. "But what my discoveries have shown is that in fact even very small programs can be quite capable of doing all sorts of sophisticated computations." This point, often repeated as if it is brand new, is not new. To have to read such material is distracting at best, and adds nothing to the presentation.

There is also a lack of acknowledgment of the work of others. The main text contains virtually no mention of anyone other than "I" (Wolfram). There are no references to any other papers or books, although some researchers are mentioned by name in the voluminous notes at the end; Wolfram suggests that those interested can easily do a search and find source material on any concepts he mentions. While it is understandable that one might not wish to get too detailed in compiling references in a work as broad as this, it does make following up on the thoughts more difficult. Perhaps at least references to key papers and books could be given, especially for topics that may not be well-known.

These two stylistic issues detract from the presentation of the material. Science depends for its progress on an open exchange of ideas and fair credit where it is due, along with links to related work. Also, in our commercial age when most people are trying to sell you something, exaggerated-seeming claims automatically put up protective filters for most people. The material and ideas in this book are interesting enough to stand on their own merits as a worthwhile (if not yet truly revolutionary) synthesis.

Now, what about the science in the book? One clear flaw is the ongoing conclusion from many of the experiments that the program output seems "remarkably complex". Well, how complex is "remarkably complex"? In Chapter 10 Wolfram discusses this issue to some extent, but not satisfactorily in my opinion.

Ben Goertzel has a worthwhile take on the insufficiency of computation universality for a system to be interesting. The other reviews listed in the references section also mention various flaws from different points of view. Thompson's book is an example of material which overlaps with some of the claims in the book, with a higher density of usable mathematical and algorithmic content.

My feeling is that Wolfram could have benefited from more interaction with others and peer-review. (Given his financial resources, he could for instance have been creative and (say) funded a couple of dozen grad students every year on a part-time basis to suggest connections and similar work in other fields, previous material, and so on.) A nice quote from Anders Sandberg on the power of parallel researching:

"While academia is a kind of kindergarten and full of distractions, the continual interaction with other researchers is a very powerful intelligence amplification tool: you get criticisms, ideas and data from a large number of parallel researchers. Even if you are a genius the law of comparative advantage implies that you are better off trading information with others rather than working on your own."

5 Suggestions, Conclusions, and References

Suggestions for a 2nd edition:

- More links to the rest of science.
- Less ego.
- Be a little more careful in making claims.
- Beef up the main text with the notes and related material.
- Think carefully about what "complex" means...

I would also say that the existence of this book is an implicit suggestion for the computational complexity theory community to write more interpretive surveys of its own work for a broad audience, and perhaps to connect more to other areas and "big questions". Aside from extracting the usable models and suggestions from the book, perhaps the theory community can regard *A New Kind of Science* as a challenge to do better?

Wolfram has written a clear book which asks a lot of interesting questions, but it's not a breakthrough yet. However, it may still be worth going through for those interested in exploring the power of simple computational models.

References

Scott Aaronson's review (focused on computational complexity and physics): arxiv.org/abs/quant-ph/0206089

Edwin Clark's list of reviews: http://www.math.usf.edu/~eclark/ANKOS_reviews.html

Ben Goertzel's review: www.extropy.org/ideas/journal/current/2002-06-01.html

Ray Kurzweil's review: www.kurzweilai.net/articles/art0464.html?printable=1

James Thompson, 2001. *Simulation: A Modeler's Approach*.

Wolfram's site for the book: www.wolframscience.com