JOURNAL OF NUMERICAL ANALYSIS AND APPROXIMATION THEORY

J. Numer. Anal. Approx. Theory, vol. 54 (2025) no. 1, pp. 195–196,

ictp.acad.ro/jnaat

BOOK REVIEWS

Peter M. Kogge, *The Zen of Exotic Computing*, SIAM, Philadelphia, 2023, XXXI + 386 pp., ISBN 978-1-61197-728-8 (paperback), eISBN 978-1-61197-729-5.

Written by one of the more experienced architects of digital systems, this book proposes an unusual exercise: to step back from the machine we all use and ask, over and over again, what computing actually is. Peter M. Kogge is the Ted H. McCourtney Professor of Computer Science and Engineering at the University of Notre Dame and an IBM Fellow; he is the inventor of the Kogge–Stone adder, the designer of the Space Shuttle I/O processor and of EXECUBE, one of the earliest processor-in-memory chips. The motivation he states in the Preface is that we may be nearing the end of an eighty-year run of one of the most consequential metaphors in the history of technology—the von Neumann model—and that, with the slowing of Moore's law, alternative models of computation are becoming increasingly significant. The book is his attempt to describe a large cross section of those models, old and new, in a common language and with a common terminology.

The volume is organized in eight parts covering twenty-three chapters, followed by a bibliography and a very carefully prepared index. Each chapter after the background material is devoted to a single model of computation and is deliberately kept short—the author aimed at ten to twenty pages—so that what is essential about each model is conveyed without extra detail. The chapters are largely self-contained, and every one of them closes with a *For further reading* section that serves as a starting point for a deeper study of the respective topic.

Parts I and II are introductory. The first two chapters discuss what is meant by computation, by levels of abstraction and by computer architecture, and then lay out the spectrum along which the later models will be placed: imperative versus declarative, discrete versus continuous, value-based versus symbolic, deterministic versus nondeterministic. Chapters 3 and 4 are a brisk review of the mathematics that recurs throughout the book: sets, relations, functions, languages and regular expressions, graphs, Markov chains and probability distributions, followed by a chapter on linear algebra which also introduces the Dirac bra-ket notation, Hilbert spaces, and quadratic unconstrained boolean optimization.

Part III treats what the author calls the classical model. Finite state automata, pushdown automata, Turing machines and their many variants are presented in a compact but formal manner, and Chapter 7 collects the limits of computing: complexity classes, reducibility, NP-completeness, undecidability and the halting problem, together with an extended discussion of boolean satisfiability. Part III closes by deriving the von Neumann and Harvard architectures from a universal Turing machine, in a sequence of small and convincing steps.

The remaining parts leave the standard model behind. Part IV is devoted to intrinsically parallel computation: Flynn's taxonomy, pipelining and out-of-order execution, PRAM and MIMD models, memory consistency and atomicity, the bulk synchronous parallel model, MPI, Amdahl's law and strong versus weak scaling; then cellular automata, systolic arrays, Petri nets and MapReduce. The systolic chapter will be of particular interest to this journal's readers, as it works through FFT and bitonic sort arrays, polynomial evaluation, convolution, matrix multiplication and matrix–vector products, and discusses how multidimensional recurrences may be mapped onto such arrays. Part V turns to symbolic alternatives—lambda calculus, combinators, and logic computing with its inference engines, Horn clauses and Herbrand universes—where computation is rewriting rather than stepping. Part VI presents models built on a different mathematical basis: probabilistic Turing machines, Las Vegas and Monte Carlo algorithms, simulated annealing and the probabilistic complexity classes, followed by reversible computing, reversible gates and the billiard ball computer.

The last two parts are the ones that give the book its title. Part VII covers models inspired by physical phenomena: neuromorphic computing, from natural neurons to artificial neural nets, supervised and unsupervised learning, back propagation and spiking networks; and analog electrical models, where operational amplifiers and integrators are used to solve differential equations. Part VIII covers models that can only be carried out by using the phenomena themselves: DNA computing, including Adleman's original experiment and DNA-based SAT solving; optical and photonic computing, with spectral transforms, 2D convolution and holographic storage; and finally an ample chapter on quantum computing, which takes the reader from qubits, entanglement and the Bloch sphere through the gate model and the no-cloning theorem to the quantum Fourier transform, Shor's and Grover's algorithms, and quantum annealing.

A thread that runs through the whole book, and holds it together, is the recurring question of how each model compares in power to the Turing machine. Almost every chapter returns to it, so that the reader acquires not a catalogue of curiosities but a map on which every model has a position relative to a fixed reference point.

The text was designed for a one-semester course at an advanced undergraduate or beginning graduate level, and the Preface contains a diagram with several suggested paths through the material, adapted to students with a computer science, an automata theory, or an electrical engineering and physics background. Since the number of interesting models proved larger than a single semester can hold, the author split them between a basic set, which forms this printed volume, and a bonus set available on the publisher's website, together with problems, projects and lists of online resources for each chapter; the online index uses a chapter-page format, so that additions to one chapter do not disturb the references to the others. A background in computer science is useful but not indispensable—some exposure to linear algebra is the more important prerequisite.

The book is a pleasant and stimulating read, and it is rare to find such breadth handled with this economy of means. It is warmly recommended to students and teachers in computer science and applied mathematics, and to numerical analysts curious about the machines on which their algorithms may run in the coming decades. Readers looking for a deep treatment of any one of these models will need the further reading the author carefully provides, but as an organized introduction to the whole menagerie, this volume has few competitors.

Imre Boros