Algorithmic Adventures: From Knowledge to Magic*

Review by Antonio E. Porreca†
porreca@disco.unimib.it

January 15, 2010

1 Introduction

Algorithmic Adventures: From Knowledge to Magic by Juraj Hromkovič is an introductory book to theoretical computer science. The author describes some of the most important areas of the discipline (and some less basic ones) while keeping the use of formalism and mathematics to a minimum whenever possible, and focuses on unexpected and fascinating results. The main goal of the author is to show that computer science is not a dull, purely technical subject but instead, quoting his own words, it is *full of magic*.

2 Summary

The first chapter of the book, *A short story about the development of computer science or why computer science is not a computer driving licence*, begins by listing some of the common misconceptions about the discipline, particularly arguing against the identification of computer science with a set of ICT skills. The author then discusses the importance of *definitions* and *axioms* (in the informal sense of "principles") for science, and of *implication* in proofs. The search for a procedure for deciding mathematical truths, as proposed by David Hilbert, is cited as the origin of the concept of *algorithm*, hence of computer science itself.

The second chapter *Algorithmics, or what have programming and baking in common?* describes what an algorithm is by using the classic recipe metaphor, which is analysed in detail in order to show the differences between the two concepts. The importance of the choice of the basic operations allowed is stressed.

*Juraj Hromkovič, *Algorithmic Adventures: From Knowledge to Magic*, Springer, 2009. 363 pages, 41.55 €. Online version available at http://dx.doi.org/10.1007/978-3-540-85986-4.

†Dipartimento di Informatica, Sistemistica e Comunicazione, Università degli Studi di Milano-Bicocca. Viale Sarca 336/14, 20126 Milano, Italy.

Programming is then described as the activity of implementing algorithms; a simple programming language, together with a few complete programs, is presented for this purpose.

The third chapter, *Infinity is not equal to infinity*, covers enough basic set theory to be able to prove that different infinite cardinalities exist, as shown by Cantor's diagonal argument. Here Hromkovič discusses extensively why the existence of bijections (which he calls "matchings") between sets is the right way to compare their sizes, instead of a more naive approach based on the subset relation, which fails for infinite sets.

This discussion about cardinality has the consequence, described in the fourth chapter *Limits of computability*, that there exist uncomputable real numbers (i.e., their decimal expansion is not the output of any program). Once again by diagonal argument, a concrete undecidable set is then described. Reductions between problems are also introduced, together with their use in proving existence or non-existence of algorithms (in particular, the undecidability of the halting problem is proved).

The fifth chapter is titled *Complexity theory or what to do when the energy of the universe doesn't suffice for performing a computation?*. Here the author explains that decidability of a problem is usually not enough, and that *efficient* solutions are needed. Then he introduces a simple time complexity measure for programs and analyses some examples from this point of view. The importance of the notion of polynomial time is discussed, and the notion of NP-completeness is informally described. As an example, the author describes a polynomial time (Karp) reduction of the vertex cover problem to the problem of checking the existence of Boolean solutions to a set of inequalities. As a remedy against NP-hardness, approximation algorithms are introduced via the standard 2-approximation for vertex cover.

Chapter six, *Randomness in nature and as a source of efficiency in algorithms*, begins with a discussion about the historical views on randomness. Then, the author shows how randomness can be exploited to obtain efficient solutions where deterministic ones are impossible. The example is a $O(\log n)$ communication complexity randomised protocol for deciding equality, which can be repeated several times in order to exponentially decrease the probability of error. No deterministic protocol for equality has sublinear communication complexity.

Cryptography, or how to transform drawbacks into advantages, the seventh chapter of the book, introduces "a magical science of the present time". After a brief historical excursion, including Kerckhoffs' principle, the author describes one-time pad encryption and why it is often impractical although unbreakable. Protocols for secret sharing are then introduced, via the usual padlock metaphor. The author shows that these *cannot* be implemented in a secure way by using the XOR operation with two secret keys (a simple example of cryptanalysis); the Diffie-Hellman protocol is presented as a working solution. Public-key cryptosystems based on one-way functions are informally described, together with digital signatures.

The last three chapters discuss relatively recent areas of computer science, or areas which are unusual for an introductory book. The eighth one is *Computing with DNA molecules, or biological computer technology on the horizon*. Here Hromkovič describes the structure of DNA, how it can be used to encode information and which basic operations (useful from a computational standpoint) can be performed in a laboratory setting. Finally, he describes Adleman's experiment, which provided a concrete implementation of an algorithm for finding Hamiltonian paths in graphs.

The ninth chapter *Quantum computers, or computing in the Wonderland of particles* informally describes some counter-intuitive properties of particles, the double-slit experiment, how qubits work in terms of probability amplitudes and how they collapse into a classic state when measured. The operations on qubits are described as unitary matrix transformations. Due to the modest mathematical background assumed, no concrete quantum algorithm is described.

The final chapter, *How to make good decisions for an unknown future or how to foil an adversary*, deals with online algorithms. These are algorithms which receive their input in pieces during the execution, instead of having it all available before starting. Since they need to choose a strategy based on partial information, an adversary (a model of the worst-case input) may exploit this weakness by making the algorithm perform poorly. A computer memory paging problem is shown not to possess online algorithms with a good competitiveness (i.e., the ratio between the cost of the solution it finds and the cost of the optimal solution); once again, randomness is shown to be useful in finding a good solution with high probability, for the problem of allocating work stations in a factory to a multi-step task.

3 Opinion

I think the author is successful in identifying a set of intriguing topics, which clearly show that computer science is full of surprises. In particular, undecidability, which arose during the very first steps of the discipline itself, and cryptography are well-known sources of "magic" results. Of course, the book is not meant to be a comprehensive catalogue of all aspects of computer science, not even as far as the more theoretical side is concerned, but the selection of topics is ample enough, and likely to light a spark of interest in the mathematically-minded readers.

The ideal audience of *Algorithmic Adventures* probably consists of undergraduate students with an interest in learning the fundamentals of computer science; the later chapters of the book feel a bit too technical for the general public, particularly since the number of references provided is limited.

I completely agree with the author on the necessity to explain what computer science is really about, and on his enthusiasm about the "magic" results which have been obtained (an enthusiasm which clearly shows through in his

writing), and I believe this book is a step in the right direction. I really liked the presence of a chapter on set theory, which alone proves the existence of unsolvable problems, and I find that several portions of this books may also be useful to graduate students or researchers trying to familiarise themselves with an unknown subject: for instance, having only a vague understanding of communication complexity, I found the description of the randomised protocol for equality an excellent introduction. The single chapters of the book, which are also available in electronic format, might also be used as introductory reading material for courses on the specific topics.

Although my overall opinion is positive, I found some negative points about this book. The bibliography is very limited, and some topics are presented without references; hence, the reader interested in learning more about set theory, computability or complexity is mostly left on his own. The chapter on quantum computing is not very successful in conveying the fundamental ideas about the topic, which probably require a lot more space and a more mathematical and physical treatment in order to be tackled effectively. Finally, I found the historical and philosophical remarks scattered around the chapters a bit sketchy and imprecise. Nonetheless, this is a good "tourist guidebook" for those who want to take a deeper look at computer science.