Undergraduate Topics in Computer Science

Undergraduate Topics in Computer Science (UTiCS) delivers high-quality instructional content for undergraduates studying in all areas of computing and information science. From core foundational and theoretical material to final-year topics and applications, UTiCS books take a fresh, concise, and modern approach and are ideal for self-study or for a one- or two-semester course. The texts are all authored by established experts in their fields, reviewed by an international advisory board, and contain numerous examples and problems. Many include fully worked solutions.

For further volumes:
http://www.springer.com/series/7592

David Makinson

Sets, Logic and Maths for Computing

Second edition

Springer

David Makinson
London School of Economics
Department of Philosophy
Houghton Street
WC2A 2AE London
United Kingdom
david.makinson@gmail.com

ISSN 1863-7310
ISBN 978-1-4471-2499-3 e-ISBN 978-1-4471-2500-6
DOI 10.1007/978-1-4471-2500-6
Springer London Dordrecht Heidelberg New York

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

Library of Congress Control Number: 2012930652

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

The first part of this preface is for the student, the second for the instructor. Some readers may find it helpful to look at both. Whoever you are, welcome!

For the Student

You have finished secondary school and are about to begin at a university or technical college. You want to study computing. The course includes some mathematics – and that was not necessarily your favourite subject. But there is no escape: a certain amount of finite mathematics is a required part of the first-year curriculum, because it is a necessary toolkit for the subject itself.

What Is in This Book?

That is where this book comes in. Its purpose is to provide the basic mathematical language that you need to enter the world of the information and computer sciences.

It does not contain all the mathematics that you will need through the several years of your undergraduate career. There are other very good, quite massive, volumes that do that. At some stage, you will find it useful to get one and keep it on your shelf for reference. But experience has convinced this author that no matter how good the compendia are, beginning students tend to feel intimidated, lost and unclear about what parts to focus on. This short book, in contrast, offers just the basics that you need to know from the beginning, on which you can build further as needed.

It also recognizes that you may not have done much mathematics at school, may not have understood very well what was going on and may even have grown to detest it. No matter: you can learn the essentials here and perhaps even have fun doing so.

So what is the book about? It is about certain tools that we need to apply over and over again when thinking about computations. They include:

1. *Collecting* things together. In the jargon of mathematics, this is *set theory*.
2. *Comparing* things. This is the theory of *relations*.
3. *Associating* one item with another. Introduces the key notion of a *function*.
4. *Recycling outputs as inputs*. We introduce the ideas of *recursion* and *induction*.

5. *Counting*. The mathematician calls it *combinatorics*.
6. *Weighing the odds*. This is done with *probability*.
7. *Squirrel math*. Here we make use of *trees*.
8. *Yea and Nay*. Just two truth-values underlie *propositional logic*.
9. *Everything and nothing*. That is what *quantificational logic* is about.
10. *Just supposing*. How *complex proofs* are built out of simple ones.

Without an understanding of these basic concepts, large portions of computer science remain behind closed doors. As you begin to grasp the ideas and integrate them into your thought, you will also find that their application extends far beyond computing into many other areas.

How Should You Use It?

The good news is that there is not all that much to commit to memory. Your sister studying medicine, or brother going for law, will envy you terribly for this. In our subject, the essential things are to *understand* and be able to *apply*.

But that is a much more subtle affair than one might imagine, as the two are interdependent. Application without understanding is blind and quickly leads to errors – often trivial, but sometimes ghastly. On the other hand, comprehension remains poor without the practice given by applications. In particular, you do not fully understand a definition until you have seen how it takes effect in specific situations: positive examples reveal its scope, negative ones show its limits. It also takes some experience to be able to recognize *when* you have really understood something, compared to having done no more than recite the words or call upon them in hope of blessing.

For this reason, exercises have been included as an indispensable part of the learning process. Skip them at your peril: no matter how simple and straightforward a concept seems, you will not fully assimilate it unless you apply it. That is part of what is meant by the old proverb ‘there is no royal road in mathematics’. Even when a problem is accompanied by a sample solution, you will benefit much more if you first hide the answer and try to work it out for yourself. That requires self-discipline, but it brings real rewards. Moreover, the exercises have been chosen so that in many instances the result is just what is needed to make a step somewhere later in the book. They are thus integral to the development of the general theory.

By the same token, do not get into the habit of skipping proofs when you read the text. Postpone, yes, but omit, no. In mathematics, you have never fully understood a fact unless you have also grasped *why* it is true, that is, have assimilated at least one proof of it. The well-meaning idea that mathematics can be democratized by teaching the ‘facts’ and forgetting about the proofs wrought disaster in some secondary and university education systems.

In practice, the mathematical tools that we bulleted above are rarely applied in isolation from each other. They gain their real power when used in concert, setting up a crossfire that can bring tough problems to the ground. For example, the concept of a set, once explained in the first chapter, is used everywhere in what follows. Relations reappear in graphs, trees and logic. Functions are omnipresent.

For the Instructor

Any book of this kind needs to find delicate balances between competing virtues and shortcomings of modes of presentation and choices of material. We explain the vision behind the choices made here.

Manner of Presentation

Mathematically, the most elegant and coherent way to proceed is to begin with the most general concepts and gradually unfold them so that the more specific and familiar ones appear as special cases. Pedagogically, this sometimes works, but it can also be disastrous. There are situations where the reverse is often required: begin with some of the more familiar examples and special cases, and then show how they may naturally be broadened.

There is no perfect solution to this problem; we have tried to find a least imperfect one. Insofar as we begin the book with sets, relations and functions in that order, we are following the first path. But, in some chapters, we have followed the second one. For example, when explaining induction and recursion, we begin with the most familiar special case, simple induction/recursion over the positive integers; then pass to their cumulative forms over the same domain; broaden to their qualitatively formulated structural versions; and finally present the most general forms on arbitrary well-founded sets. Again, in the chapter on trees, we have taken the rather unusual step of beginning with rooted trees, where intuition is strongest and applications abound, then abstracting to unrooted trees.

In the chapters on counting and probability, we have had to strike another balance – between traditional terminology and notation antedating the modern era and its translation into the language of sets, relations and functions. Most textbook presentations do it all in the traditional way, which has its drawbacks. It leaves the student in the dark about the relation of this material to what was taught in earlier chapters on sets and functions. And, frankly, it is not always very rigorous or transparent. Our policy is to familiarize the reader with *both* kinds of presentation – using the language of sets and functions for a clear understanding of the material itself and the traditional languages of combinatorics and probability to permit communication in the local dialect.

In those two chapters, yet another balance had to be found. One can easily supply counting formulae and probability equalities to be committed to memory and applied in drills. It is more difficult to provide reader-friendly explanations and proofs that permit students to understand what they are doing and why. This book tries to do both, with a rather deeper commitment to the latter than is usual. In particular, it is emphasized that whenever we wish to count the number of selections of k items from a pool of n, a definite answer is possible only when it is clearly understood whether the selections admit repetition and whether they discriminate between orders, giving four options and thus four different counting formulae for the toolbox. The student must learn which tool to choose, and why, as much as how to use it.

The place of logic in the story is delicate. We have left its systematic exposition to the end – a decision that may seem rather strange, as one uses logic whenever reasoning mathematically, even about the most elementary things discussed in the first chapters. Do we not need a chapter on logic at the very beginning of the book? The author's experience in the classroom tells him that in practice that does not work well. Despite its simplicity – perhaps indeed because of it – logic can be elusive for beginning students. It acquires intuitive content only as examples of its employment are revealed. Moreover, it turns out that a really clear explanation of the basic concepts of logic requires some familiarity with the mathematical notions of sets, relations, functions and trees.

For these reasons, the book takes a different tack. In its early chapters, notions of logic are identified briefly as they arise in the discussion of more 'concrete' material. This is done in 'logic boxes'. Each box introduces just enough to get on with the task in hand. Much later, in the last three chapters, all this is brought together and extended. By then, the student should have little trouble appreciating what the subject is all about and how natural it all is, and will be ready to use other basic mathematical tools when studying it.

From time to time, there are boxes of a different nature – 'Alice boxes'. This little troublemaker comes from the pages of Lewis Carroll to ask embarrassing questions in all innocence. Often they are on points that students find puzzling, but which they have trouble articulating clearly or are too shy to pose. It is hoped that the Mad Hatter's replies are of assistance to them as well as to Alice.

The house of mathematics can be built in many different ways, and students often have difficulty reconciling the formulations and constructions of one text with those of another. From time to time, we comment on such differences. Two examples, which always give trouble if not dealt with explicitly, arise in quantificational (first-order) logic. They concern different, although ultimately equivalent, ways of reading the quantifiers and different ways of using the terms 'true' and 'false' for formulae of containing free variables.

Choice of Topics

Overall, our choice of topics is fairly standard, as the chapter titles indicate. If strapped for class time, an instructor could omit some of the starred sections of Chaps. 4, 5, 6, 7, 8, 9, 10. But it is urged that Chaps. 1, 2, 3 be taught entire, as everything in them is useful to the computer science student and is also needed in following chapters. When sufficient classroom time is available, some teachers may, on the other hand, wish to deepen topics or add further ones.

We have not included a chapter on the theory of graphs. That was a difficult call to make, and the reasons were as follows. Although trees are a particular kind of graph, there is no difficulty in covering everything we want to say about trees without entering into general graph theory. Moreover, an adequate treatment of graphs, even if squeezed into one chapter of about the same length as the others, takes a good 2 weeks of additional class time to cover properly with enough examples and exercises to make it sink in. The basic theory of graphs is a rather messy topic, with a rather high definition/theorem ratio and multiple options about how wide to cast the net (directed/undirected graphs, with or without loops, multiple edges and so on). The author's experience is that students gain little from a high-speed run through a series of distinctions and definitions, memorized for the examinations and then promptly forgotten.

On the other hand, recursion and induction are here developed in more detail than is usual in texts of this kind, where it is common to neglect recursive definition in favour of inductive proof and to restrict attention to the natural numbers. Although Chap. 4 begins with the natural numbers, it goes on to explain number-free forms, in particular the often neglected structural ones that are so important for computer science and logic. We also explain the general versions of induction and recursion for arbitrary well-founded relations. Throughout the presentation, the interplay between recursive definition and inductive proof is brought out, with the latter piggybacking on the former. This chapter ends up being the longest in the book.

Finally, a decision had to be made whether to include specific algorithms and, if so, in what form: ordinary English, pseudo-code outline or a real-life programming language in full detail? Most first-year students of computing will be taking, in parallel, courses on principles of programming and some specific programming language; but the languages chosen differ from one institution to another and change over time. The policy in this text is to sketch the essential idea of basic algorithms in plain but carefully formulated English. In a few cases (particularly the chapter on trees), we give optional exercises in expressing them in pseudo-code. Instructors wishing to make more systematic use of pseudo-code, or to link material with specific programming languages, should feel free to do so.

Courses Outside Computer Science

Computer science students are not the only ones who need to learn about these topics. Students of mathematics, philosophy as well as the more theoretical sides of linguistics, economics and political science, all need a course in basic formal methods covering more or less the same territory. This text can be used, with some omissions and additions, for a formal methods course in any of those disciplines.

In the particular case of philosophy, in the second half of the twentieth century, there was an unfortunate tendency to teach only elementary logic, leaving aside instruction on sets, relations, functions, recursion/induction, probability and trees. Even within logic, the focus was rather narrow, often almost entirely on so-called natural deduction. But as already remarked, it is difficult for the student to get a clear idea of what is going on in logic, and difficult for the instructor to say anything interesting about it, without having those other concepts available in the toolkit. The few philosophy students going on to more advanced courses in logic are usually exposed to such tools in bits and pieces but without a systematic grounding. It is the author's belief that all of the subjects dealt with in this book (with the exception of Chap. 5 on counting and the last two sections of Chap. 7 on trees) are also vital for an adequate course on formal methods for students of philosophy. With some additional practice on the ins and outs of approximating statements of ordinary language in the notation of propositional and predicate logic, the book can also be used as a text for such a course.

The Second Edition

For those already familiar with the first edition, we mention the modifications made for the second one. On a general level, affecting the entire book, the main developments are as follows:

- Formulations have been reviewed and retouched to improve clarity and simplicity. Sometimes the order of development within a chapter has been changed slightly. In the end, hardly a paragraph remains untouched.
- Further exercises have been added at points where classroom experience suggested that they would be useful.
- Additional sample answers have also been provided. But, resisting classroom pressures, not all exercises are supplied with answers! After students have learned to swim in shallow water, they need to acquire confidence and self-reliance by venturing out where their feet no longer touch bottom. Ultimately that leads to the satisfaction of getting things right where no crib is possible. Instructors can also use these exercises in tests and examinations.
- To facilitate cross-reference, the numbering of exercises has been aligned with that of the sections and subsections. Thus, Exercise *x.y.z* (*k*) is the *k*th exercise in Subsection *x.y.z* of Chapter *x*, with the (*k*) omitted when there is only one exercise for the subsection. When a section has no subsections, the numbering is reduced to *x.y* (*k*), and end-of-chapter exercises are listed simply as *x* (*k*), where *x* is the chapter number.

On the level of specific chapters, there have been significant revisions in the content and presentation of Chaps. 5, 6, 7:

- In Chap. 5 on counting, the section on rearrangements and configured partitions has been rewritten to make it clearer and more reader-friendly, and some material of lesser interest has been removed.
- In Chap. 6 on probability, the last sections have been refashioned, and some material of little interest in the finite context has been eliminated.
- In Chap. 7 on trees, we have clarified the presence and role of the empty tree, which had been left rather vague in the first edition.

The chapters on propositional and first-order logic have been very much reorganized, emerging as three rather than two chapters:

- *Order of presentation*. The general notion of a consequence (or closure) relation has been taken from its place at very beginning of Chap. 8 and relocated in the new Chap. 10. The previous position was appropriate from a theoretical point of view, but from a pedagogical one it just did not work. So the abstract concept has been kept under wraps until readers have become thoroughly familiar with the particular cases of classical propositional and first-order consequence.
- *Motivation*. Moreover, the concept of a consequence relation is now motivated by showing how it supports the validity of 'elementary derivations', understood as finite sequences or trees of propositions made up of individually valid steps. It is shown that the conditions defining consequence relations are just what are needed for elementary proofs to do their job.
- *Content*. The sections of Chaps. 8 and 9 that in the first edition were devoted to the skill of constructing 'natural deductions' in the symbols of propositional and quantificational logic have been suppressed. In their place, the new Chap. 10 includes an *informal explanation* of the main strategies of mathematical proof, *rigorous statements* of the higher-level rules on which those strategies are based and an explanation on how the rules support traditional practice. Roughly speaking, a formal drill of dubious value has given way to greater attention to a good understanding of the recursive structure of proofs.
- *Other*. The notation used in explaining the semantics of quantificational logic in Chap. 9 has been considerably streamlined, to keep it as simple as possible without loss of clarity or rigour.

A section of the author's webpage http://sites.google.com/site/davidcmakinson is set aside for material relating to this edition: residual errata, comments, further exercises, etc. Readers are invited to send their observations to david.makinson@gmail.com.

Acknowledgements

The author would like to thank Anatoli Degtyarev, Franz Dietrich, Valentin Goranko and George Kourousias for helpful discussions when writing the first edition. George also helped prepare the diagrams. Thanks to the London School of Economics (LSE) and successive department heads Colin Howson, Richard Bradley and Luc Bovens for the wonderful working environment that they provided while both editions were being written. Herbert Huber's emails helped find first-edition typos. For the second edition, Abhaya Nayak and Xavier Parent made valuable comments on a draft of the new tenth chapter.

Saint Martin de Hinx

Contents

List of Figures