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 of which include fully worked solutions.

The UTiCS concept relies on high-quality, concise books in softback format, and generally a maximum of 275–300 pages. For undergraduate textbooks that are likely to be longer, more expository, Springer continues to offer the highly regarded Texts in Computer Science series, to which we refer potential authors.

More information about this series at http://www.springer.com/series/7592

David Makinson

Sets, Logic and Maths for Computing

Third Edition

Springer

David Makinson
Department of Philosophy,
Logic and Scientific Method
London School of Economics
London, UK

ISSN 1863-7310 ISSN 2197-1781 (electronic)
Undergraduate Topics in Computer Science
ISBN 978-3-030-42217-2 ISBN 978-3-030-42218-9 (eBook)
https://doi.org/10.1007/978-3-030-42218-9

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

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 informatics or 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 required for entering the world of the information and computing 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, often 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 a compendium is, beginning students tend to feel intimidated, lost, and unclear about what is in it 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, from the world of sets,

- *Collecting* things together. In the jargon of mathematics, first steps with *sets*.
- *Comparing* things. This is the theory of *relations*.
- *Associating* one item with another. Introduces the key notion of a *function*.
- *Recycling outputs as inputs*. We explain the ideas of *recursion* and *induction*.

From other parts of finite mathematics,

- *Counting*. The mathematician calls it *combinatorics*.
- *Weighing the odds*. This is done with *probability*.
- *Squirrel math*. Here we make use of *trees*.

From logic,

- *Yea and Nay*. Just two truth-values underlie *propositional logic*.
- *Everything and nothing*. That is what *quantificational logic* is about.
- *Just supposing*. How *complex proofs* are built out of simple ones.
- *Sticking to the Point*. How to make logic sensitive to relevance (new to this edition).

How Should You Use It?

Without an understanding of 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. So, there is work to be done.

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 two essential things are to *understand* and to be able to *apply*.

But that is a 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 register 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 integral part of the learning process. Skip them at your peril. That is part of what is meant by the old proverb 'there is no royal road in mathematics'. Although all exercises in this edition are accompanied by a solution, you will benefit much more if you first cover 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. Thus, they are also part of the development of the general theory.

By the same token, don't get into the habit of skipping verifications when you read the text. Postpone, yes, but omit, no. In mathematics, you have never fully appreciated a fact unless you have also grasped *why* it is true, i.e. 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 has wrought disaster in some secondary and university education systems.

In practice, the tools that are 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 the competing virtues and shortcomings in different choices of material and ways of presenting it.

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 often it is disastrous. There are situations where the reverse is 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 minimally 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 for the same domain; broaden to their qualitatively formulated structural versions; finally, give the most general articulation, 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 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 approach is to familiarize readers 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 counting and probability to permit rapid 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. Again, 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 should learn which tool to choose when, and why, as well 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. Don't we 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 meaning 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 early chapters, notions of logic are identified briefly as they arise in the discussion and verification 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 four 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 to help study it.

From time to time there are boxes of a different nature—'Alice boxes'. This little trouble-maker 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. Quite often, we comment on such variations. In particular, two points in quantificational (first-order) logic always give trouble if variants are not compared explicitly. One concerns distinct, although ultimately equivalent, ways of reading the quantifiers; the other arises from differing conventions for using the terms 'true' and 'false' in application to formulae containing free variables.

Choice of Topics

Overall, our choice of topics is fairly standard, as can be seen from the chapter titles. If strapped for class time, an instructor can omit some later sections of some chapters, or even entire chapters that are not close to the interests of the students or the dictates of the curriculum. For example, few computer science curricula will require the material of Chaps. 10 and 11 (levels of inference, logic with relevance) and few philosophy students will be fired up by the Chap. 5 (counting) or the last part of Chap. 7 (on unrooted trees). But it is urged that Chaps. 1–3, the first four sections of Chap. 7, as well as 8 and 9, be taught uncut as just about everything in them is useful to everybody.

We have not included a chapter on the theory of graphs. That was a difficult call to make, and the reasons for the decision were as follows. Although trees are a particular kind of graph, there is no difficulty in covering everything we want to say about rooted 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 two 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 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 induction on the natural numbers it goes on to explain number-free forms of both inductive proof and recursive definition including, in particular the often-neglected structural forms that are so important for computer science, logic and theoretical linguistics. We also explain the very general versions based on arbitrary well-founded relations. Throughout the presentation, the interplay between recursive definition and inductive proof is brought out, with the latter piggy-backing on the former. This chapter ends up being the longest in the book.

Instructors may be surprised that the chapters on logic do not attempt to drill readers in building derivations in formal notation, following what is known as a system of 'natural deduction'. The reason is that the author, after years teaching a variety of such systems, has become convinced that they are of marginal pedagogical benefit. Students take a long time to become accustomed to the idiosyncrasies of whichever layout they are exposed to, tend to lose sight of essential principles in the finicky procedural details, and forget most of it as soon as the exams are over. Instead, Chap. 10, on proof and consequence, seeks to make clear the basic ideas that underlie natural deduction, explaining the difference between first-level, second-level and split-level derivations as well as how to squeeze a derivation tree into a derivation sequence and how to flatten a split-level derivation into a format that mimics a first-level one.

Finally, a decision had to be made whether to include specific algorithms in the book. Most first-year students of computing will be taking courses, in parallel, 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. Instructors wishing 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 to master 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 case of philosophy there was, in the second half of the twentieth century, an unfortunate tendency to teach only elementary logic, leaving aside any instruction on sets, relations, functions, recursion/induction, probability and trees. The few students going on to more advanced courses in logic were usually exposed to such tools in bits and pieces, but without a systematic grounding. Even within logic, the election of material was often quite narrow, focussing almost entirely on natural deduction. But as already remarked, it is difficult for the student to get a clear idea of what is going on in logic without having those other concepts available in the tool-kit. 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 unrooted trees) are equally vital for an adequate course for students of philosophy. With some additional practice on the subtle art of making approximate representations of statements from ordinary language in the language of propositional and predicate logic, the book can also be used as a text for such a course.

The Third Edition

The first edition of this book was published in 2008; the second appeared in 2012 and the many changes made were itemized in the preface for that edition. This third edition adds Chap. 11, *Sticking to the point: relevance in logic*, which introduces procedures of syntactic control for logical consequence; it should be good fun for students of computing as well as those from philosophy.

In previous editions, only about half of the exercises were accompanied by a solution and some readers using the text for self-study let the author know their unhappiness with that situation. This edition provides solutions to all exercises, mostly in full but occasionally in outline. It is hoped that, as a result, the text is friendlier than before to readers working alone and takes some of the burden off instructors. To be honest, the task of writing out solutions in full also helped better appreciate the student's situation facing the exercises, leading to many modifications. Together, the solutions and new chapter increase the size of the book by about a third.

Even more than in the first two editions, the third takes the time to inform readers of shorthand ways of speaking and useful 'abuses of language'. Such information is more important than often realized. In the author's experience, when a student has trouble assimilating a point, as often as not it is a result of misunderstanding the language in which it is made.

The notation chosen for the text is the same as in the previous editions except for two items, both in the chapters on logic. For classical logical consequence, we now use the standard double turnstile $\vDash$ rather than $\vdash$, reserving the latter sign for consequence relations in general. For the substitution of a term t for all free occurrences of a variable x in a formula α we now write $\alpha_{x:=t}$ instead of the more common notation $\alpha(t/x)$ used in the previous editions; this is to ensure that there can be no hesitation about which way round the substitution goes when t is itself a variable.

To bring out its general structure, the book has now been divided into three broad parts: *Sets* (including mathematical objects built with them, such as relations and functions), *Math* (in the traditional quantitative sense focusing on counting and finite probability), and *Logic*. Of course, this partition leaves open the question where to put the chapter on induction and recursion, which begins with induction on the natural numbers while going on to general issues of structural and well-founded recursion and induction, as also the chapter on trees which, conversely, begins as part of the theory of relations but ends with a combinatorial flavour. The former was put in *Sets*, while the latter went into *Math*, but the reverse allocation would be almost as appropriate.

As the third edition was being prepared a colleague asked whether, leaving aside all options of presentation, the book has content that is not easily found in other introductory texts. The author's response is relayed here for anyone with the same question. Sections 4.6 (structural recursion and induction), 4.7.3 (defining functions by well-founded recursion on their domains) and 8.4.4 (most modular versions of sets of propositional formulae) bring within the grasp of a beginner important notions that are usually left unexplored. The same is true of most of Chap. 10 (consequence relations and higher-order rules of inference) as well as all of Chap. 11 (relevance in logic).

A section of the author's webpage https://sites.google.com/site/davidcmakinson/ is earmarked for material relating to this edition: typos and other errata, additional comments, further exercises, etc. As the text goes to press these rubrics are empty but, with the help of readers, they will surely soon be populated. Observations should go to david.makinson@gmail.com.

London, UK David Makinson
February 2020

Acknowledgements The late Colin Howson and ensuing department heads at the London School of Economics (LSE) provided a wonderful working environment and the opportunity to test material in classes. Anatoli Degtyarev, Franz Dietrich, Valentin Goranko, George Kourousias, Abhaya Nayak and Xavier Parent provided helpful comments on drafts of the first two editions; George also helped with the diagrams. For the third edition, Chap. 11 benefitted from valuable remarks by Lloyd Humberstone, Abhaya Nayak, Xavier Parent, Nick Smith and Peter Smith.

A number of readers and LSE students drew attention to typos that remained in the first two editions: Luc Batty, Alex Bendig, Michael Broschat, Panagiotis Dimakis, Niklas Fors, Rick Greer, Herbert Huber, Deng (Joe) Jingyuan, Regina Lai, Daniel Mak, Pascal Meier, Fredrik Nyberg, Daniel Shlomo. Thanks to you all.

Contents

Part III Logic