Distributed Computing *through* Combinatorial Topology

MK
MORGAN KAUFMANN

Maurice Herlihy
Dmitry Kozlov
Sergio Rajsbaum

Distributed Computing Through Combinatorial Topology

Maurice Herlihy

Dmitry Kozlov

Sergio Rajsbaum

ELSEVIER

AMSTERDAM • BOSTON • HEIDELBERG • LONDON
NEW YORK • OXFORD • PARIS • SAN DIEGO
SAN FRANCISCO • SINGAPORE • SYDNEY • TOKYO

Morgan Kaufmann is an imprint of Elsevier

MK MORGAN KAUFMANN

Acquiring Editor: *Todd Green*
Editorial Project Manager: *Lindsay Lawrence*
Project Manager: *Punithavathy Govindaradjane*
Designer: *Maria Inês Cruz*

Morgan Kaufmann is an imprint of Elsevier
225 Wyman Street, Waltham, MA 02451, USA

Library of Congress Cataloging-in-Publication Data
Herlihy, Maurice.
Distributed computing through combinatorial topology / Maurice Herlihy, Dmitry Kozlov, Sergio Rajsbaum.
pages cm
Includes bibliographical references and index.
ISBN 978-0-12-404578-1 (alk. paper)
1. Electronic data processing–Distributed processing–Mathematics.
2. Combinatorial topology. I. Kozlov, D. N. (Dmitrii Nikolaevich) II. Rajsbaum, Sergio. III. Title.
QA76.9.D5H473 2013
004'.36–dc23

2013038781

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

ISBN: 978-0-12-404578-1
Printed and bound in the United States of America
14 15 16 17 18 10 9 8 7 6 5 4 3 2 1

Distributed Computing Through Combinatorial Topology

For my parents, David and Patricia Herlihy, and for Liuba, David, and Anna.
To Esther, David, Judith, and Eva-Maria.
Dedicated to the memory of my grandparents, Itke and David, Anga and Sigmund, and to the memory of my Ph.D. advisor, Shimon Even.

Acknowledgments

We thank all the students, colleagues, and friends who helped improve this book: Hagit Attiya, Irina Calciu, Armando Castañeda, Lisbeth Fajstrup, Eli Gafni, Eric Goubault, Rachid Guerraoui, Damien Imbs, Petr Kuznetsov, Hammurabi Mendez, Yoram Moses, Martin Raussen, Michel Raynal, David Rosenblueth, Ami Paz, Vikram Seraph, Nir Shavit, Christine Tasson, Corentin Travers, and Mark R. Tuttle. We apologize for any names inadvertently omitted.

Special thanks to Eli Gafni for his many insights on the algorithmic aspects of this book.

Preface

This book is intended to serve as a textbook for an undergraduate or graduate course in theoretical distributed computing or as a reference for researchers who are, or want to become, active in this area. Previously, the material covered here was scattered across a collection of conference and journal publications, often terse and using different notations and terminology. Here we have assembled a self-contained explanation of the mathematics for computer science readers and of the computer science for mathematics readers.

Each of these chapters includes exercises. We think it is essential for readers to spend time solving these problems. Readers should have some familiarity with basic discrete mathematics, including induction, sets, graphs, and continuous maps. We have also included mathematical notes addressed to readers who want to explore the deeper mathematical structures behind this material.

The first three chapters cover the *fundamentals* of combinatorial topology and how it helps us understand distributed computing. Although the mathematical notions underlying our computational models are elementary, some notions of combinatorial topology, such as simplices, simplicial complexes, and levels of connectivity, may be unfamiliar to readers with a background in computer science. We explain these notions from first principles, starting in Chapter 1, where we provide an intuitive introduction to the new approach developed in the book. In Chapter 2 we describe the approach in more detail for the case of a system consisting of two processes only. Elementary graph theory, which is well-known to both computer scientists and mathematicians, is the only mathematics needed.

The graph theoretic notions of Chapter 2 are essentially one-dimensional simplicial complexes, and they provide a smooth introduction to Chapter 3, where most of the topological notions used in the book are presented. Though similar material can be found in many topology texts, our treatment here is different. In most texts, the notions needed to model computation are typically intermingled with a substantial body of other material, and it can be difficult for beginners to extract relevant notions from the rest. Readers with a background in combinatorial topology may want to skim this chapter to review concepts and notations.

The next four chapters are intended to form the core of an advanced undergraduate course in distributed computing. The mathematical framework is self-contained in the sense that all concepts used in this section are defined in the first three chapters.

In this part of the book we concentrate on the so-called *colorless* tasks, a large class of coordination problems that have received a great deal of attention in the research literature. In Chapter 4, we describe our basic operational and combinatorial models of computation. We define tasks and asynchronous, fault-tolerant, wait-free shared-memory protocols. This chapter explains how the mathematical language of combinatorial topology (such as simplicial complexes and maps) can be used to describe concurrent computation and to identify the colorless tasks that can be solved by these protocols. In Chapter 5, we apply these mathematical tools to study colorless task solvability by more powerful protocols. We first consider computational models in which processes fail by *crashing* (unexpectedly halting). We give necessary and sufficient conditions for solving colorless tasks in a range of different computational models, encompassing different crash-failure models and different forms of communication. In Chapter 6, we show how the same mathematical notions can be extended to deal with *Byzantine* failures, where faulty processes, instead of crashing, can display arbitrary behavior. In Chapter 7, we show how to use *reductions* to transform results about one model of computation to results about others.

Chapters 8–11 are intended to form the core of a graduate course. Here, too, the mathematical framework is self-contained, although we expect a slightly higher level of mathematical sophistication. In this part, we turn our attention to *general tasks*, a broader class of problems than the colorless tasks covered earlier. In Chapter 8, we describe how the mathematical framework previously used to model colorless tasks can be generalized, and in Chapter 9 we consider manifold tasks, a subclass of tasks with a particularly nice geometric structure. We state and prove Sperner's lemma for manifolds and use this to derive a separation result showing that some problems are inherently "harder" than others. In Chapter 10, we focus on how computation affects *connectivity*, informally described as the question of whether the combinatorial structures that model computations have "holes." We treat connectivity in an axiomatic way, avoiding the need to make explicit mention of homology or homotopy groups. In Chapter 11, we put these pieces together to give necessary and sufficient conditions for solving general tasks in various models of computation. Here notions from elementary point-set topology, such as open covers and compactness are used.

The final part of the book provides an opportunity to delve into more advanced topics of distributed computing by using further notions from topology. These chapters can be read in any order, mostly after having studied Chapter 8. Chapter 12 examines the renaming task, and uses combinatorial theorems such as the Index Lemma to derive lower bounds on this task. Chapter 13 uses the notion of shellability to show that a number of models of computation that appear to be quite distinct can be analyzed with the same formal tools. Chapter 14 examines simulations and reductions for general tasks, showing that the shared-memory models used interchangeably in this book really are equivalent. Chapter 15 draws a connection between a certain class of tasks and the Word Problem for finitely-presented groups, giving a hint of the richness of the universe of tasks that are studied in distributed computing. Finally, Chapter 16 uses Schlegel diagrams to prove basic topological properties about our core models of computation.

Maurice Herlihy was supported by NSF grant 000830491.

Sergio Rajsbaum by UNAM PAPIIT and PAPIME Grants.

Dmitry Kozlov was supported by the University of Bremen and the German Science Foundation.

Companion Site

This book offers complete code for all the examples, as well as slides, updates, and other useful tools on its companion web page at:
https://store.elsevier.com/product.jsp?isbn=9780124045781&pagename=search.

CONTENTS

Acknowledgments xi
Preface xiii

PART I FUNDAMENTALS

CHAPTER 1 Introduction 3
1.1 Concurrency Everywhere 3
1.2 Distributed Computing 9
1.3 Two Classic Distributed Computing Problems 12
1.4 Chapter Notes 18
1.5 Exercises 19

CHAPTER 2 Two-Process Systems 21
2.1 Elementary Graph Theory 22
2.2 Tasks 25
2.3 Models of Computation 28
2.4 Approximate Agreement 33
2.5 Two-Process Task Solvability 36
2.6 Chapter Notes 37
2.7 Exercises 38

CHAPTER 3 Elements of Combinatorial Topology 41
3.1 Basic Concepts 42
3.2 Simplicial Complexes 44
3.3 Standard Constructions 47
3.4 Carrier Maps 50
3.5 Connectivity 53
3.6 Subdivisions 55
3.7 Simplicial and Continuous Approximations 60
3.8 Chapter Notes 64
3.9 Exercises 64

PART II COLORLESS TASKS

CHAPTER 4 Colorless Wait-Free Computation 69
4.1 Operational Model 70
4.2 Combinatorial Model 78

4.3 The Computational Power of Wait-Free Colorless Immediate Snapshots 88
4.4 Chapter Notes 92
4.5 Exercises 93

CHAPTER 5 Solvability of Colorless Tasks in Different Models 97
5.1 Overview of Models 98
5.2 t-Resilient Layered Snapshot Protocols 99
5.3 Layered Snapshots with k-Set Agreement 103
5.4 Adversaries 105
5.5 Message-Passing Protocols 108
5.6 Decidability 112
5.7 Chapter Notes 117
5.8 Exercises 118

CHAPTER 6 Byzantine-Resilient Colorless Computation 123
6.1 Byzantine Failures 123
6.2 Byzantine Communication Abstractions 125
6.3 Byzantine Set Agreement 128
6.4 Byzantine Barycentric Agreement 128
6.5 Byzantine Task Solvability 129
6.6 Byzantine Shared Memory 131
6.7 Chapter Notes 132
6.8 Exercises 132

CHAPTER 7 Simulations and Reductions 135
7.1 Motivation 135
7.2 Combinatorial Setting 137
7.3 Applications 139
7.4 BG Simulation 140
7.5 Conclusions 143
7.6 Chapter Notes 144
7.7 Exercises 145

PART III GENERAL TASKS

CHAPTER 8 Read-Write Protocols for General Tasks 149
8.1 Overview 149
8.2 Tasks 150
8.3 Examples of Tasks 152
8.4 Protocols 158

8.5 Chapter Notes 163
8.6 Exercises 164

CHAPTER 9 Manifold Protocols 167
9.1 Manifold Protocols 168
9.2 Layered Immediate Snapshot Protocols 173
9.3 No Set Agreement from Manifold Protocols 178
9.4 Set Agreement vs. Weak Symmetry Breaking 182
9.5 Chapter Notes 188
9.6 Exercises 189

CHAPTER 10 Connectivity 191
10.1 Consensus and Path Connectivity 191
10.2 Immediate Snapshot Model and Connectivity 193
10.3 k-Set Agreement and $(k - 1)$ Connectivity 199
10.4 Immediate Snapshot Model and k-Connectivity 199
10.5 Chapter Notes 203
10.6 Exercises 204

CHAPTER 11 Wait-Free Computability for General Tasks 205
11.1 Inherently Colored Tasks: The Hourglass Task 205
11.2 Solvability for Colored Tasks 208
11.3 Algorithm Implies Map 212
11.4 Map Implies Algorithm 212
11.5 A Sufficient Topological Condition 222
11.6 Chapter Notes 227
11.7 Exercises 227

PART IV ADVANCED TOPICS

CHAPTER 12 Renaming and Oriented Manifolds 231
12.1 An Upper Bound: Renaming with $2n + 1$ Names 232
12.2 Weak Symmetry Breaking 236
12.3 The Index Lemma 237
12.4 Binary Colorings 240
12.5 A Lower Bound for $2n$-Renaming 242
12.6 Chapter Notes 244
12.7 Exercises 244

CHAPTER 13 Task Solvability in Different Models **247**
13.1 Shellability 248
13.2 Examples 249
13.3 Pseudospheres 251
13.4 Carrier Maps and Shellable Complexes 253
13.5 Applications 256
13.6 Chapter Notes 270
13.7 Exercises 271

CHAPTER 14 Simulations and Reductions for Colored Tasks **273**
14.1 Model 273
14.2 Shared-Memory Models 275
14.3 Trivial Reductions 277
14.4 Layered Snapshot from Read-Write 277
14.5 Immediate Snapshot from Snapshot 279
14.6 Immediate Snapshot from Layered Immediate Snapshot 280
14.7 Snapshot from Layered Snapshot 284
14.8 Exercises 287
14.9 Chapter Notes 287

CHAPTER 15 Classifying Loop Agreement Tasks **289**
15.1 The Fundamental Group 290
15.2 Algebraic Signatures 291
15.3 Main Theorem 293
15.4 Applications 296
15.5 Torsion Classes 297
15.6 Conclusions 297
15.7 Chapter Notes 297
15.8 Exercises 298

CHAPTER 16 Immediate Snapshot Subdivisions **299**
16.1 A Glimpse of Discrete Geometry 299
16.2 Chapter Notes 304
16.3 Exercises 304

Bibliography **305**

Index **313**

PART I

Fundamentals