

Neil Immerman

DESCRIPTIVE COMPLEXITY

With 41 Illustrations



Springer

Neil Immerman
Department of Computer Science
University of Massachusetts
Amherst, MA 01003-4610, USA

Series Editors

David Gries
Fred B. Schneider

Department of Computer Science
Cornell University
Upson Hall
Ithaca, NY 14853-7501, USA

Library of Congress Cataloging-in-Publication Data
Immerman, Neil, 1953–

Descriptive complexity / Neil Immerman.

p. cm. — (Graduate texts in computer science)

Includes bibliographical references and index.

ISBN 978-1-4612-6809-3 ISBN 978-1-4612-0539-5 (eBook)

DOI 10.1007/978-1-4612-0539-5

1. Computational complexity. 2. Logic, Symbolic and mathematical.

I. Title. II. Series: Graduate texts in computer science (Springer-Verlag New York Inc.)

QA267.7.I46 1998

511.3—dc21

98-33563

Printed on acid-free paper.

© 1999 Springer Science+Business Media New York

Originally published by Springer-Verlag New York Berlin Heidelberg in 1999

Softcover reprint of the hardcover 1st edition 1999

All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the publisher (Springer Science+Business Media, LLC), except for brief excerpts in connection with reviews or scholarly analysis. Use in connection with any form of information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed is forbidden.

The use of general descriptive names, trade names, trademarks, etc., in this publication, even if the former are not especially identified, is not to be taken as a sign that such names, as understood by the Trade Marks and Merchandise Marks Act, may accordingly be used freely by anyone.

Production managed by Allan Abrams; manufacturing supervised by Jacqui Ashri.

Photocomposed copy prepared from the author's LaTeX files.

9 8 7 6 5 4 3 2 1

ISBN 978-1-4612-6809-3

SPIN 10688630

Contents

Preface	vii
Introduction	1
1 Background in Logic	5
1.1 Introduction and Preliminary Definitions	5
1.2 Ordering and Arithmetic	12
1.2.1 $\text{FO}(\text{BIT}) = \text{FO}(\text{PLUS}, \text{TIMES})$	14
1.3 Isomorphism	16
1.4 First-Order Queries	17
2 Background in Complexity	23
2.1 Introduction	23
2.2 Preliminary Definitions	24
2.3 Reductions and Complete Problems	27
2.4 Alternation	34
2.5 Simultaneous Resource Classes	40
2.6 Summary	41
3 First-Order Reductions	45
3.1 $\text{FO} \subseteq \text{L}$	45
3.2 Dual of a First-Order Query	46
3.3 Complete problems for L and NL	50
3.4 Complete Problems for P	53

4	Inductive Definitions	57
4.1	Least Fixed Point	57
4.2	The Depth of Inductive Definitions	61
4.3	Iterating First-Order Formulas	63
5	Parallelism	67
5.1	Concurrent Random Access Machines	68
5.2	Inductive Depth Equals Parallel Time	70
5.3	Number of Variables Versus Number of Processors	74
5.4	Circuit Complexity	77
5.5	Alternating Complexity	85
5.5.1	Alternation as Parallelism	87
6	Ehrenfeucht-Fraïssé Games	91
6.1	Definition of the Games	91
6.2	Methodology for First-Order Expressibility	99
6.3	First-Order Properties Are Local	102
6.4	Bounded Variable Languages	104
6.5	Zero-One Laws	107
6.6	Ehrenfeucht-Fraïssé Games with Ordering	109
7	Second-Order Logic and Fagin's Theorem	113
7.1	Second-Order Logic	113
7.2	Proof of Fagin's Theorem	115
7.3	NP-Complete Problems	119
7.4	The Polynomial-Time Hierarchy	121
8	Second-Order Lower Bounds	125
8.1	Second-Order Games	125
8.2	$\text{SO}\exists(\text{monadic})$ Lower Bound on Reachability	129
8.3	Lower Bounds Including Ordering	133
9	Complementation and Transitive Closure	139
9.1	Normal Form Theorem for $\text{FO}(\text{LFP})$	139
9.2	Transitive Closure Operators	143
9.3	Normal Form for $\text{FO}(\text{TC})$	144
9.4	Logspace is Primitive Recursive	148
9.5	$\text{NSPACE}[s(n)] = \text{co-NSPACE}[s(n)]$	149
9.6	Restrictions of SO	151
10	Polynomial Space	157
10.1	Complete Problems for PSPACE	157
10.2	Partial Fixed Points	160
10.3	$\text{DSPACE}[n^k] = \text{VAR}[k + 1]$	162
10.4	Using Second-Order Logic to Capture PSPACE	165

11 Uniformity and Precomputation	169
11.1 An Unbounded Number of Variables	170
11.1.1 Tradeoffs Between Variables and Quantifier Depth . . .	171
11.2 First-Order Projections	171
11.3 Help Bits	176
11.4 Generalized Quantifiers	177
12 The Role of Ordering	181
12.1 Using Logic to Characterize Graphs	182
12.2 Characterizing Graphs Using $\mathcal{L}^k$	183
12.3 Adding Counting to First-Order Logic	185
12.4 Pebble Games for $\mathcal{C}^k$	187
12.5 Vertex Refinement Corresponds to $\mathcal{C}^2$	189
12.6 Abiteboul-Vianu and Otto Theorems	193
12.7 Toward a Language for Order-Independent P	199
13 Lower Bounds	203
13.1 Håstad's Switching Lemma	203
13.2 A Lower Bound for REACH_a	208
13.3 Lower Bound for Fixed Point and Counting	213
14 Applications	221
14.1 Databases	221
14.1.1 SQL	222
14.1.2 Datalog	224
14.2 Dynamic Complexity	226
14.2.1 Dynamic Complexity Classes	227
14.3 Model Checking	234
14.3.1 Temporal Logic	235
14.4 Summary	239
15 Conclusions and Future Directions	241
15.1 Languages That Capture Complexity Classes	241
15.1.1 Complexity on the Face of a Query	243
15.1.2 Stepwise Refinement	244
15.2 Why Is <i>Finite</i> Model Theory Appropriate?	244
15.3 Deep Mathematical Problems: P versus NP	245
15.4 Toward Proving Lower Bounds	246
15.4.1 Role of Ordering	246
15.4.2 Approximation and Approximability	247
15.5 Applications of Descriptive Complexity	248
15.5.1 Dynamic Complexity	248
15.5.2 Model Checking	248
15.5.3 Abstract State Machines	248
15.6 Software Crisis and Opportunity	249

15.6.1 How can Finite Model Theory Help?	250
References	251
Index	263