

Discrete

Kenneth H. Rosen

Mathematics and Its Applications

SEVENTH EDITION

Discrete Mathematics and Its

Applications

Seventh Edition

Kenneth H. Rosen

*Monmouth University
(and formerly AT&T Laboratories)*



DISCRETE MATHEMATICS AND ITS APPLICATIONS, SEVENTH EDITION

Published by McGraw-Hill, a business unit of The McGraw-Hill Companies, Inc., 1221 Avenue of the Americas, New York, NY 10020. Copyright © 2012 by The McGraw-Hill Companies, Inc. All rights reserved. Previous editions © 2007, 2003, and 1999. No part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written consent of The McGraw-Hill Companies, Inc., including, but not limited to, in any network or other electronic storage or transmission, or broadcast for distance learning.

Some ancillaries, including electronic and print components, may not be available to customers outside the United States.

This book is printed on acid-free paper.

ISBN 978-0-07-338309-5
MHID 0-07-338309-0

Vice President & Editor-in-Chief: *Marty Lange*
Editorial Director: *Michael Lange*
Global Publisher: *Raghothaman Srinivasan*
Executive Editor: *Bill Stenquist*
Development Editors: *Lorraine K. Buczek/Rose Kernan*
Senior Marketing Manager: *Curt Reynolds*
Project Manager: *Robin A. Reed*
Buyer: *Sandy Ludovissy*
Design Coordinator: *Brenda A. Rolwes*
Cover painting: Jasper Johns, *Between the Clock and the Bed*, 1981. Oil on Canvas (72 × 126 1/4 inches) Collection of the artist. Photograph by Glenn Stiegelman. Cover Art © Jasper Johns/Licensed by VAGA, New York, NY Cover Designer: *Studio Montage, St. Louis, Missouri*
Lead Photo Research Coordinator: *Carrie K. Burger*
Media Project Manager: *Tammy Juran*
Production Services/Compositor: *RPK Editorial Services/PreTeX, Inc.*
Typeface: *10.5/12 Times Roman*
Printer: *R.R. Donnelley*

All credits appearing on this page or at the end of the book are considered to be an extension of the copyright

page. Library of Congress Cataloging-in-Publication Data

Rosen, Kenneth H.
Discrete mathematics and its applications / Kenneth H. Rosen. — 7th ed.
p. cm.
Includes index.
ISBN 0–07–338309–0
1. Mathematics. 2. Computer science—Mathematics. I. Title.
QA39.3.R67 2012
511–dc22

2011011060

www.mhhe.com

Contents

About the Author vi
Preface vii
The Companion Website xvi
To the Student xvii

1 The Foundations: Logic and Proofs	1
1.1 Propositional Logic	1
1.2 Applications of Propositional Logic	16
1.3 Propositional Equivalences	25
1.4 Predicates and Quantifiers	36
1.5 Nested Quantifiers	57
1.6 Rules of Inference	69
1.7 Introduction to Proofs	80
1.8 Proof Methods and Strategy	92
End-of-Chapter Material	109
2 Basic Structures: Sets, Functions, Sequences, Sums, and Matrices	115

2.1 Sets	115
2.2 Set Operations	127
2.3 Functions	138
2.4 Sequences and Summations	156
2.5 Cardinality of Sets	170
2.6 Matrices	177
<i>End-of-Chapter Material</i>	185

3 Algorithms 191

3.1 Algorithms	191
3.2 The Growth of Functions	204
3.3 Complexity of Algorithms	218
<i>End-of-Chapter Material</i>	232

4 Number Theory and Cryptography 237

4.1 Divisibility and Modular Arithmetic	237
4.2 Integer Representations and Algorithms	245
4.3 Primes and Greatest Common Divisors	257
4.4 Solving Congruences	274
4.5 Applications of Congruences	287
4.6 Cryptography	294
<i>End-of-Chapter Material</i>	306

iii

iv Contents

5 Induction and Recursion 311

5.1 Mathematical Induction	311
5.2 Strong Induction and Well-Ordering	333
5.3 Recursive Definitions and Structural Induction	344
5.4 Recursive Algorithms	360
5.5 Program Correctness	372
<i>End-of-Chapter Material</i>	377

6 Counting 385

6.1 The Basics of Counting	385
6.2 The Pigeonhole Principle	399
6.3 Permutations and Combinations	407
6.4 Binomial Coefficients and Identities	415
6.5 Generalized Permutations and Combinations	423
6.6 Generating Permutations and Combinations	434
<i>End-of-Chapter Material</i>	439

7 Discrete Probability	445
7.1 An Introduction to Discrete Probability	445
7.2 Probability Theory	452
7.3 Bayes' Theorem	468
7.4 Expected Value and Variance	477
<i>End-of-Chapter Material</i>	494

8 Advanced Counting Techniques	501
8.1 Applications of Recurrence Relations	501
8.2 Solving Linear Recurrence Relations	514
8.3 Divide-and-Conquer Algorithms and Recurrence Relations	527
8.4 Generating Functions	537
8.5 Inclusion–Exclusion	552
8.6 Applications of Inclusion–Exclusion	558
<i>End-of-Chapter Material</i>	565

9 Relations	573
9.1 Relations and Their Properties	573
9.2 n -ary Relations and Their Applications	583
9.3 Representing Relations	591
9.4 Closures of Relations	597
9.5 Equivalence Relations	607
9.6 Partial Orderings	618
<i>End-of-Chapter Material</i>	633
	Contents v

10 Graphs	641
10.1 Graphs and Graph Models	641
10.2 Graph Terminology and Special Types of Graphs	651
10.3 Representing Graphs and Graph Isomorphism	668
10.4 Connectivity	678
10.5 Euler and Hamilton Paths	693
10.6 Shortest-Path Problems	707
10.7 Planar Graphs	718
10.8 Graph Coloring	727
<i>End-of-Chapter Material</i>	735

11 Trees	745
11.1 Introduction to Trees	745
11.2 Applications of Trees	757
11.3 Tree Traversal	772
11.4 Spanning Trees	

..... 785	11.5 Minimum Spanning Trees	803
..... 797	<i>End-of-Chapter Material</i>	
	803
12 Boolean Algebra		811
12.1 Boolean Functions		
... 811	12.2 Representing Boolean Functions	
..... 819	12.3 Logic Gates	
..... 822	12.4 Minimization of Circuits	
..... 828	<i>End-of-Chapter Material</i>	
	843
13 Modeling Computation		847
13.1 Languages and Grammars		
.. 847	13.2 Finite-State Machines with Output	
..... 858	13.3 Finite-State Machines with No Output	
..... 865	13.4 Language Recognition	
..... 878	13.5 Turing Machines	
..... 888	<i>End-of-Chapter Material</i>	
	899
Appendixes		A-1
1 Axioms for the Real Numbers and the Positive Integers	1	2 Exponential and Logarithmic Functions
..... 7	3 Pseudocode	
.....	11	
<i>Suggested Readings B-1</i>		
<i>Answers to Odd-Numbered Exercises S-1</i>		
<i>Photo Credits C-1</i>		
<i>Index of Biographies I-1</i>		
<i>Index I-2</i>		

About the Author

Kenneth H. Rosen has had a long career as a Distinguished Member of the Technical Staff

at AT&T Laboratories in Monmouth County, New Jersey. He currently holds the position of Visiting Research Professor at Monmouth University, where he teaches graduate courses in computer science.

Dr. Rosen received his B.S. in Mathematics from the University of Michigan, Ann Arbor (1972), and his Ph.D. in Mathematics from M.I.T. (1976), where he wrote his thesis in the area of number theory under the direction of Harold Stark. Before joining Bell Laboratories in 1982, he held positions at the University of Colorado, Boulder; The Ohio State University, Columbus; and the University of Maine, Orono, where he was an associate professor of mathematics. While working at AT&T Labs, he taught at Monmouth University, teaching courses in discrete mathematics, coding theory, and data security. He currently teaches courses in algorithm design and in computer security and cryptography.

Dr. Rosen has published numerous articles in professional journals in number theory and in mathematical modeling. He is the author of the widely used *Elementary Number Theory*

and *Its Applications*, published by Pearson, currently in its sixth edition, which has been translated into Chinese. He is also the author of *Discrete Mathematics and Its Applications*, published by McGraw-Hill, currently in its seventh edition. *Discrete Mathematics and Its Applications* has sold more than 350,000 copies in North America during its lifetime, and hundreds of thousands of copies throughout the rest of the world. This book has also been translated into Spanish, French, Greek, Chinese, Vietnamese, and Korean. He is also co-author of *UNIX: The Complete Reference*; *UNIX SystemV Release 4: An Introduction*; and *Best UNIX Tips Ever*, all published by Osborne McGraw-Hill. These books have sold more than 150,000 copies, with translations into Chinese, German, Spanish, and Italian. Dr. Rosen is also the editor of the *Handbook of Discrete and Combinatorial Mathematics*, published by CRC Press, and he is the advisory editor of the CRC series of books in discrete mathematics, consisting of more than 55 volumes on different aspects of discrete mathematics, most of which are introduced in this book. Dr. Rosen serves as an Associate Editor for the journal *Discrete Mathematics*, where he works with submitted papers in several areas of discrete mathematics, including graph theory, enumeration, and number theory. He is also interested in integrating mathematical software into the educational and professional environments, and worked on several projects with Waterloo Maple Inc.'s MapleTM software in both these areas. Dr. Rosen has also worked with several publishing companies on their homework delivery platforms.

At Bell Laboratories and AT&T Laboratories, Dr. Rosen worked on a wide range of projects, including operations research studies, product line planning for computers and data communications equipment, and technology assessment. He helped plan AT&T's products and services in the area of multimedia, including video communications, speech recognition, speech synthesis, and image networking. He evaluated new technology for use by AT&T and did standards work in the area of image networking. He also invented many new services, and holds more than 55 patents. One of his more interesting projects involved helping evaluate technology for the AT&T attraction that was part of EPCOT Center.

Preface

In writing this book, I was guided by my long-standing experience and interest in teaching

discrete mathematics. For the student, my purpose was to present material in a precise, readable manner, with the concepts and techniques of discrete mathematics clearly presented and demonstrated. My goal was to show the relevance and practicality of discrete mathematics to students, who are often skeptical. I wanted to give students studying computer science all of the mathematical foundations they need for their future studies. I wanted to give mathematics students an understanding of important mathematical concepts together with a sense of why these concepts are important for applications. And most importantly, I wanted to accomplish these goals without watering down the material.

For the instructor, my purpose was to design a flexible, comprehensive teaching tool using proven pedagogical techniques in mathematics. I wanted to provide instructors with a package of materials that they could use to teach discrete mathematics effectively and efficiently in the most appropriate manner for their particular set of students. I hope that I have achieved these goals.

I have been extremely gratified by the tremendous success of this text. The many improve

ments in the seventh edition have been made possible by the feedback and suggestions of a large number of instructors and students at many of the more than 600 North American schools, and at any many universities in parts of the world, where this book has been successfully used.

This text is designed for a one- or two-term introductory discrete mathematics course taken by students in a wide variety of majors, including mathematics, computer science, and engineering. College algebra is the only explicit prerequisite, although a certain degree of mathematical maturity is needed to study discrete mathematics in a meaningful way. This book has been designed to meet the needs of almost all types of introductory discrete mathematics courses. It is highly flexible and extremely comprehensive. The book is designed not only to be a successful textbook, but also to serve as valuable resource students can consult throughout their studies and professional life.

Goals of a Discrete Mathematics Course

A discrete mathematics course has more than one purpose. Students should learn a particular set of mathematical facts and how to apply them; more importantly, such a course should teach students how to think logically and mathematically. To achieve these goals, this text stresses mathematical reasoning and the different ways problems are solved. Five important themes are interwoven in this text: mathematical reasoning, combinatorial analysis, discrete structures, algorithmic thinking, and applications and modeling. A successful discrete mathematics course should carefully blend and balance all five themes.

1. *Mathematical Reasoning:* Students must understand mathematical reasoning in order to read, comprehend, and construct mathematical arguments. This text starts with a discussion of mathematical logic, which serves as the foundation for the subsequent discussions of methods of proof. Both the science and the art of constructing proofs are addressed. The technique of mathematical induction is stressed through many different types of examples of such proofs and a careful explanation of why mathematical induction is a valid proof technique.

2. *Combinatorial Analysis:* An important problem-solving skill is the ability to count or enumerate objects. The discussion of enumeration in this book begins with the basic techniques of counting. The stress is on performing combinatorial analysis to solve counting problems and analyze algorithms, not on applying formulae.
3. *Discrete Structures:* A course in discrete mathematics should teach students how to work with discrete structures, which are the abstract mathematical structures used to represent discrete objects and relationships between these objects. These discrete structures include sets, permutations, relations, graphs, trees, and finite-state machines.
4. *Algorithmic Thinking:* Certain classes of problems are solved by the specification of an algorithm. After an algorithm has been described, a computer program can be constructed implementing it. The mathematical portions of this activity, which include the specification of the algorithm, the verification that it works properly, and the analysis of the computer memory and time required to perform it, are all covered in this text. Algorithms are described using both English and an easily understood form of pseudocode.
5. *Applications and Modeling:* Discrete mathematics has applications to almost every conceivable area of study. There are many applications to computer science and data networking in this text, as well as applications to such diverse areas as chemistry, biology, linguistics, geography, business, and the Internet. These applications are natural and important uses of discrete mathematics and are not contrived. Modeling with discrete mathematics is an extremely important problem-solving skill, which students have the opportunity to develop

by constructing their own models in some of the exercises.

Changes in the Seventh Edition

Although the sixth edition has been an extremely effective text, many instructors, including longtime users, have requested changes designed to make this book more effective. I have devoted a significant amount of time and energy to satisfy their requests and I have worked hard to find my own ways to make the book more effective and more compelling to students.

The seventh edition is a major revision, with changes based on input from more than 40 formal reviewers, feedback from students and instructors, and author insights. The result is a new edition that offers an improved organization of topics making the book a more effective teaching tool. Substantial enhancements to the material devoted to logic, algorithms, number theory, and graph theory make this book more flexible and comprehensive. Numerous changes in the seventh edition have been designed to help students more easily learn the material. Additional explanations and examples have been added to clarify material where students often have difficulty. New exercises, both routine and challenging, have been added. Highly relevant applications, including many related to the Internet, to computer science, and to mathematical biology, have been added. The companion website has benefited from extensive development activity and now provides tools students can use to master key concepts and explore the world of discrete mathematics, and many new tools under development will be released in the year following publication of this book.

I hope that instructors will closely examine this new edition to discover how it might meet their needs. Although it is impractical to list all the changes in this edition, a brief list that highlights some key changes, listed by the benefits they provide, may be useful.

More Flexible Organization

Applications of propositional logic are found in a new dedicated section, which briefly introduces logic circuits.

Recurrence relations are now covered in Chapter 2.

Expanded coverage of countability is now found in a dedicated section in Chapter 2.

Preface ix

Separate chapters now provide expanded coverage of algorithms (Chapter 3) and number theory and cryptography (Chapter 4).

More second and third level heads have been used to break sections into smaller coherent parts.

Tools for Easier Learning

Difficult discussions and proofs have been marked with the famous Bourbaki “dangerous bend” symbol in the margin.

New marginal notes make connections, add interesting notes, and provide advice to students.

More details and added explanations, in both proofs and exposition, make it easier for students to read the book.

Many new exercises, both routine and challenging, have been added, while many existing exercises have been improved.

Enhanced Coverage of Logic, Sets, and Proof

The satisfiability problem is addressed in greater depth, with Sudoku modeled in terms of satisfiability.

Hilbert’s Grand Hotel is used to help explain uncountability.

Proofs throughout the book have been made more accessible by adding steps and reasons behind these steps.

A template for proofs by mathematical induction has been added.

The step that applies the inductive hypothesis in mathematical induction proof is now explicitly noted.

Algorithms

The pseudocode used in the book has been updated.

Explicit coverage of algorithmic paradigms, including brute force, greedy algorithms, and dynamic programming, is now provided.

Useful rules for big- O estimates of logarithms, powers, and exponential functions have been added.

Number Theory and Cryptography

Expanded coverage allows instructors to include just a little or a lot of number theory in their courses.

The relationship between the **mod** function and congruences has been explained more fully.

The sieve of Eratosthenes is now introduced earlier in the book.

Linear congruences and modular inverses are now covered in more detail.

Applications of number theory, including check digits and hash functions, are covered in great depth.

A new section on cryptography integrates previous coverage, and the notion of a cryptosystem has been introduced.

Cryptographic protocols, including digital signatures and key sharing, are now covered.

x Preface

Graph Theory

A structured introduction to graph theory applications has been added.

More coverage has been devoted to the notion of social networks.

Applications to the biological sciences and motivating applications for graph isomorphism and planarity have been added.

Matchings in bipartite graphs are now covered, including Hall's theorem and its proof.

Coverage of vertex connectivity, edge connectivity, and n -connectedness has been added, providing more insight into the connectedness of graphs.

Enrichment Material

Many biographies have been expanded and updated, and new biographies of Bellman, Bézout, Bienyamé, Cardano, Catalan, Cocks, Cook, Dirac, Hall, Hilbert, Ore, and Tao have been added.

Historical information has been added throughout the text.

Numerous updates for latest discoveries have been made.

Expanded Media

Extensive effort has been devoted to producing valuable web resources for this book.

Extra examples in key parts of the text have been provided on companion website.

Interactive algorithms have been developed, with tools for using them to explore topics

and for classroom use.

A new online ancillary, *The Virtual Discrete Mathematics Tutor*, available in fall 2012, will help students overcome problems learning discrete mathematics.

A new homework delivery system, available in fall 2012, will provide automated home work for both numerical and conceptual exercises.

Student assessment modules are available for key concepts.

Powerpoint transparencies for instructor use have been developed.

A supplement *Exploring Discrete Mathematics* has been developed, providing extensive support for using MapleTM or MathematicaTM in conjunction with the book.

An extensive collection of external web links is provided.

Features of the Book

ACCESSIBILITY This text has proved to be easily read and understood by beginning students. There are no mathematical prerequisites beyond college algebra for almost all the content of the text. Students needing extra help will find tools on the companion website for bringing their mathematical maturity up to the level of the text. The few places in the book where calculus is referred to are explicitly noted. Most students should easily understand the pseudocode used in the text to express algorithms, regardless of whether they have formally studied programming languages. There is no formal computer science prerequisite.

Each chapter begins at an easily understood and accessible level. Once basic mathematical concepts have been carefully developed, more difficult material and applications to other areas of study are presented.

Preface xi

FLEXIBILITY This text has been carefully designed for flexible use. The dependence of chapters on previous material has been minimized. Each chapter is divided into sections of approximately the same length, and each section is divided into subsections that form natural blocks of material for teaching. Instructors can easily pace their lectures using these blocks.

WRITING STYLE The writing style in this book is direct and pragmatic. Precise mathematical language is used without excessive formalism and abstraction. Care has been taken to balance the mix of notation and words in mathematical statements.

MATHEMATICAL RIGOR AND PRECISION All definitions and theorems in this text are stated extremely carefully so that students will appreciate the precision of language and rigor needed in mathematics. Proofs are motivated and developed slowly; their steps are all carefully justified. The axioms used in proofs and the basic properties that follow from them are explicitly described in an appendix, giving students a clear idea of what they can assume in a proof. Recursive definitions are explained and used extensively.

WORKED EXAMPLES Over 800 examples are used to illustrate concepts, relate different topics, and introduce applications. In most examples, a question is first posed, then its solution is presented with the appropriate amount of detail.

APPLICATIONS The applications included in this text demonstrate the utility of discrete mathematics in the solution of real-world problems. This text includes applications to a wide variety of areas, including computer science, data networking, psychology, chemistry, engineering, linguistics, biology, business, and the Internet.

ALGORITHMS Results in discrete mathematics are often expressed in terms of algorithms; hence, key algorithms are introduced in each chapter of the book. These algorithms are expressed in words and in an easily understood form of structured pseudocode, which is

described and specified in Appendix 3. The computational complexity of the algorithms in the text is also analyzed at an elementary level.

HISTORICAL INFORMATION The background of many topics is succinctly described in the text. Brief biographies of 83 mathematicians and computer scientists are included as footnotes. These biographies include information about the lives, careers, and accomplishments of these important contributors to discrete mathematics and images, when available, are displayed.

In addition, numerous historical footnotes are included that supplement the historical information in the main body of the text. Efforts have been made to keep the book up-to-date by reflecting the latest discoveries.

KEY TERMS AND RESULTS A list of key terms and results follows each chapter. The key terms include only the most important that students should learn, and not every term defined in the chapter.

EXERCISES There are over 4000 exercises in the text, with many different types of questions posed. There is an ample supply of straightforward exercises that develop basic skills, a large number of intermediate exercises, and many challenging exercises. Exercises are stated clearly and unambiguously, and all are carefully graded for level of difficulty. Exercise sets contain special discussions that develop new concepts not covered in the text, enabling students to discover new ideas through their own work.

Exercises that are somewhat more difficult than average are marked with a single star ^{*}; those that are much more challenging are marked with two stars ^{**}. Exercises whose solutions require calculus are explicitly noted. Exercises that develop results used in the text are clearly identified with the right pointing hand symbol \rightarrow . Answers or outlined solutions to all odd-

xii Preface

numbered exercises are provided at the back of the text. The solutions include proofs in which most of the steps are clearly spelled out.

REVIEW QUESTIONS A set of review questions is provided at the end of each chapter. These questions are designed to help students focus their study on the most important concepts and techniques of that chapter. To answer these questions students need to write long answers, rather than just perform calculations or give short replies.

SUPPLEMENTARY EXERCISE SETS Each chapter is followed by a rich and varied set of supplementary exercises. These exercises are generally more difficult than those in the exercise sets following the sections. The supplementary exercises reinforce the concepts of the chapter and integrate different topics more effectively.

COMPUTER PROJECTS Each chapter is followed by a set of computer projects. The approximately 150 computer projects tie together what students may have learned in computing and in discrete mathematics. Computer projects that are more difficult than average, from both a mathematical and a programming point of view, are marked with a star, and those that are extremely challenging are marked with two stars.

COMPUTATIONS AND EXPLORATIONS A set of computations and explorations is included at the conclusion of each chapter. These exercises (approximately 120 in total) are designed to be completed using existing software tools, such as programs that students or instructors have written or mathematical computation packages such as MapleTM or MathematicaTM. Many of these exercises give students the opportunity to uncover new facts and ideas through computation. (Some of these exercises are discussed in the *Exploring Discrete Mathematics* companion workbooks available online.)

WRITING PROJECTS Each chapter is followed by a set of writing projects. To do these

projects students need to consult the mathematical literature. Some of these projects are historical in nature and may involve looking up original sources. Others are designed to serve as gateways to new topics and ideas. All are designed to expose students to ideas not covered in depth in the text. These projects tie mathematical concepts together with the writing process and help expose students to possible areas for future study. (Suggested references for these projects can be found online or in the printed *Student's Solutions Guide*.)

APPENDIXES There are three appendixes to the text. The first introduces axioms for real numbers and the positive integers, and illustrates how facts are proved directly from these axioms. The second covers exponential and logarithmic functions, reviewing some basic material used heavily in the course. The third specifies the pseudocode used to describe algorithms in this text.

SUGGESTED READINGS A list of suggested readings for the overall book and for each chapter is provided after the appendixes. These suggested readings include books at or below the level of this text, more difficult books, expository articles, and articles in which discoveries in discrete mathematics were originally published. Some of these publications are classics, published many years ago, while others have been published in the last few years.

How to Use This Book

This text has been carefully written and constructed to support discrete mathematics courses at several levels and with differing foci. The following table identifies the core and optional sections. An introductory one-term course in discrete mathematics at the sophomore level can be based on the core sections of the text, with other sections covered at the discretion of the

Preface **xiii**

instructor. A two-term introductory course can include all the optional mathematics sections in addition to the core sections. A course with a strong computer science emphasis can be taught by covering some or all of the optional computer science sections. Instructors can find sample syllabi for a wide range of discrete mathematics courses and teaching suggestions for using each section of the text can be found in the *Instructor's Resource Guide* available on the website for this book.

<i>Chapter</i>	<i>Core</i>	<i>Optional CS</i>	<i>Optional Math</i>
1	1.1–1.8 (as needed)	3.1–3.3 (as	2.5
2	2.1–2.4, 2.6 (as needed)	needed) 4.5, 4.6	
3		5.4, 5.5	
4	4.1–4.4 (as needed)	6.6	
5	5.1–5.3	7.4	6.4, 6.5
6	6.1–6.3	8.3	7.2, 7.3
7	7.1	9.2	8.2, 8.4, 8.6
8	8.1, 8.5		9.4, 9.6
9	9.1, 9.3, 9.5	11.2, 11.3	10.6–10.8
10	10.1–10.5	12.1–12.4	11.4, 11.5
11	11.1	13.1–13.5	
12			
13			

Instructors using this book can adjust the level of difficulty of their course by choosing either to cover or to omit the more challenging examples at the end of sections, as well as the more challenging exercises. The chapter dependency chart shown here displays the strong dependencies. A star indicates that only relevant sections of the chapter are needed for study

of a later chapter. Weak dependencies have been ignored. More details can be found in the Instructor Resource Guide.

	Chapter 1		
		Chapter 12	Chapter 2*
Chapter 9*	Chapter	5*	
	3*	Chapter	
Chapter	Chapter	6*	
10*	4*		
Chapter 11	Chapter	Chapter 13	

manual, available separately, contains full solutions to all odd-numbered problems in the exercise sets. These solutions explain why a particular method is used and why it works. For some exercises, one or two other possible approaches are described to show that a problem can be solved in several different ways. Suggested references for the writing projects found at the end of each chapter are also included in this volume. Also included are a guide to writing proofs and an extensive description of common

Ancillaries

Chapter 7 Chapter 8

STUDENT'S SOLUTIONS GUIDE This student

xiv Preface

mistakes students make in discrete mathematics, plus sample tests and a sample crib sheet for each chapter designed to help students prepare for exams.

(ISBN-10: 0-07-735350-1) (ISBN-13: 978-0-07-735350-6)

INSTRUCTOR'S RESOURCE GUIDE This manual, available on the website and in printed form by request for instructors, contains full solutions to even-numbered exercises in the text. Suggestions on how to teach the material in each chapter of the book are provided, including the points to stress in each section and how to put the material into perspective. It also offers sample tests for each chapter and a test bank containing over 1500 exam questions to choose from. Answers to all sample tests and test bank questions are included. Finally, several sample syllabi are presented for courses with differing emphases and student ability levels.

(ISBN-10: 0-07-735349-8) (ISBN-13: 978-0-07-735349-0)

Acknowledgments

I would like to thank the many instructors and students at a variety of schools who have used this book and provided me with their valuable feedback and helpful suggestions. Their input has made this a much better book than it would have been otherwise. I especially want to thank Jerrold Grossman, Jean-Claude Evard, and Georgia Mederer for their technical reviews of the seventh edition and their "eagle eyes," which have helped ensure the accuracy of this

book. I also appreciate the help provided by all those who have submitted comments via the website.

I thank the reviewers of this seventh and the six previous editions. These reviewers have provided much helpful criticism and encouragement to me. I hope this edition lives up to their high expectations.

Reviewers for the Seventh Edition

	<i>University of Illinois, Champaign</i>	
Philip Barry	Peter Gillespie	Christopher Swanson
<i>University of Minnesota, Minneapolis</i>	<i>Fayetteville State University</i>	<i>Columbus State Community College</i>
Miklos Bona	Johannes Hattingh	Keith Olson
<i>University of Florida</i>	<i>Georgia State University</i>	<i>Utah Valley University</i>
Kirby Brown	Ken Holladay	Yongyuth Permpoontanalarp
<i>Queens College</i>	<i>University of New Orleans</i>	<i>King Mongkut's University of Technology, Thonburi</i>
John Carter	Jerry Ianni	Galin Piatniskaia
<i>University of Toronto</i>	<i>LaGuardia Community College</i>	<i>University of Missouri, St. Louis</i>
Narendra Chaudhari	Ravi Janardan	Stefan Robila
<i>Nanyang Technological University</i>	<i>University of Minnesota, Minneapolis</i>	<i>Montclair State University</i>
Allan Cochran	Norliza Katuk	<i>Auburn University</i>
<i>University of Arkansas</i>	<i>University of Utara Malaysia</i>	Sukhit Singh
Daniel Cunningham	William Klostermeyer	<i>Texas State University, San Marcos</i>
<i>Buffalo State College</i>	<i>University of North Florida</i>	David Snyder
George Davis	Przemo Kranz	<i>Texas State University, San Marcos</i>
<i>Georgia State University</i>	<i>University of Mississippi</i>	Wasin So
Andrzej Derdzinski	Jaromy Kuhl	<i>San Jose State University</i>
<i>The Ohio State University</i>	<i>University of West Florida</i>	Bogdan Suceava
Ronald Dotzel	Loredana Lanzani	<i>California State University, Fullerton</i>
<i>University of Missouri-St. Louis</i>	<i>University of Arkansas, Fayetteville</i>	<i>Ashland University</i>
T.J. Duda	Steven Leonhardi	Bon Sy
<i>Columbus State Community College</i>	<i>Winona State University</i>	<i>Queens College</i>
Bruce Elenbogen	Xu Liutong	Matthew Walsh
<i>University of Michigan, Dearborn</i>	<i>Beijing University of Posts and Telecommunications</i>	<i>Indiana-Purdue University, Fort Wayne</i>
Norma Elias	Vladimir Logvinenko	Gideon Weinstein
<i>Purdue University, Calumet-Hammond</i>	<i>De Anza Community College</i>	<i>Western Governors University</i>
Herbert Enderton	Preface xv	David Wilczynski
<i>University of California, Los Angeles</i>	Darrell Minor	<i>University of Southern California</i>
Anthony Evans		
<i>Wright State University</i>		
Kim Factor		
<i>Marquette University</i>		
Margaret Fleck	Chris Rodger	

I would like to thank Bill Stenquist, Executive Editor, for his advocacy, enthusiasm, and support. His assistance with this edition has been essential. I would also like to thank the original editor, Wayne Yuhasz, whose insights and skills helped ensure the book's success, as well as all the many other previous editors of this book.

I want to express my appreciation to the staff of RPK Editorial Services for their valuable work on this edition, including Rose Kernan, who served as both the developmental editor and the production editor, and the other members of the RPK team, Fred Dahl, Martha McMaster, Erin Wagner, Harlan James, and Shelly Gerger-Knecht. I thank Paul Mailhot of PreTeX, Inc., the compositor, for the tremendous amount of work he devoted to producing this edition, and for his intimate knowledge of LaTeX. Thanks also to Danny Meldung of Photo Affairs, Inc., who was resourceful obtaining images for the new biographical footnotes.

The accuracy and quality of this new edition owe much to Jerry Grossman and Jean-Claude Evard, who checked the entire manuscript for technical accuracy and Georgia Mederer, who checked the accuracy of the answers at the end of the book and the solutions in the *Student's Solutions Guide* and *Instructor's Resource Guide*. As usual, I cannot thank Jerry Grossman enough for all his work authoring these two essential ancillaries.

I would also express my appreciation the Science, Engineering, and Mathematics (SEM) Division of McGraw-Hill Higher Education for their valuable support for this new edition and the associated media content. In particular, thanks go to Kurt Strand: President, SEM, McGraw Hill Higher Education, Marty Lange: Editor-in-Chief, SEM, Michael Lange: Editorial Director, Raghothaman Srinivasan: Global Publisher, Bill Stenquist: Executive Editor, Curt Reynolds: Executive Marketing Manager, Robin A. Reed: Project Manager, Sandy Ludovissey: Buyer, Lorraine Buczek: In-house Developmental Editor, Brenda Rowles: Design Coordinator, Carrie K. Burger: Lead Photo Research Coordinator, and Tammy Juran: Media Project Manager.

Kenneth H. Rosen

The Companion Website

The extensive companion website accompanying this text has been substantially enhanced for the seventh edition. This website is accessible at www.mhhe.com/rosen. The homepage shows the *Information Center*, and contains login links for the site's *Student Site* and *Instructor Site*. Key features of each area are described below:

THE INFORMATION CENTER

The Information Center contains basic information about the book including the expanded table of contents (including subsection heads), the preface, descriptions of the ancillaries, and a sample chapter. It also provides a link that can be used to submit errata reports and other feedback about the book.

STUDENT SITE

The Student site contains a wealth of resources available for student use, including the following, tied into the text wherever the special icons displayed below are found in the text:

Extra Examples You can find a large number of additional examples on the site, covering all chapters of the book. These examples are concentrated in areas where students often ask for additional material. Although most of these examples amplify the basic concepts, more-challenging examples can also be found here.

Interactive Demonstration Applets These applets enable you to interactively explore how important algorithms work, and are tied directly to material in the text with linkages to examples and exercises. Additional resources are provided on how to use and apply these applets.

Self Assessments These interactive guides help you assess your understanding of 14 key concepts, providing a question bank where each question includes a brief tutorial followed by a multiple-choice question. If you select an incorrect answer, advice is provided to help you understand your error. Using these Self Assessments, you should be able to diagnose your problems and find appropriate help.

Web Resources Guide This guide provides annotated links to hundreds of external websites containing relevant material such as historical and biographical information, puzzles and problems, discussions, applets, programs, and more. These links are keyed to the text by

page number.

Additional resources in the Student site include:

Exploring Discrete Mathematics This ancillary provides help for using a computer algebra system to do a wide range of computations in discrete mathematics. Each chapter provides a description of relevant functions in the computer algebra system and how they are used, programs to carry out computations in discrete mathematics, examples, and exercises that can be worked using this computer algebra system. Two versions, *Exploring Discrete Mathematics with Maple*TM and *Exploring Discrete Mathematics with Mathematica*TM will be available.

Applications of Discrete Mathematics This ancillary contains 24 chapters—each with its own set of exercises—presenting a wide variety of interesting and important applications

covering three general areas in discrete mathematics: discrete structures, combinatorics, and graph theory. These applications are ideal for supplementing the text or for independent study.

A Guide to Proof-Writing This guide provides additional help for writing proofs, a skill that many students find difficult to master. By reading this guide at the beginning of the course and periodically thereafter when proof writing is required, you will be rewarded as your proof-writing ability grows. (Also available in the *Student's Solutions Guide*.)

Common Mistakes in Discrete Mathematics This guide includes a detailed list of common misconceptions that students of discrete mathematics often have and the kinds of errors they tend to make. You are encouraged to review this list from time to time to help avoid these common traps. (Also available in the *Student's Solutions Guide*.)

Advice on Writing Projects This guide offers helpful hints and suggestions for the Writing Projects in the text, including an extensive bibliography of helpful books and articles for research; discussion of various resources available in print and online; tips on doing library research; and suggestions on how to write well. (Also available in the *Student's Solutions Guide*.)

The Virtual Discrete Mathematics Tutor This extensive ancillary provides students with valuable assistance as they make the transition from lower-level courses to discrete mathematics. The errors students have made when studying discrete mathematics using this text has been analyzed to design this resource. Students will be able to get many of their questions answered and can overcome many obstacles via this ancillaries. The *Virtual Discrete Mathematics Tutor* is expected to be available in the fall of 2012.

INSTRUCTOR SITE

This part of the website provides access to all of the resources on the Student Site, as well as these resources for instructors:

Suggested Syllabi Detailed course outlines are shown, offering suggestions for courses with different emphases and different student backgrounds and ability levels. **Teaching Suggestions** This guide contains detailed teaching suggestions for instructors, including chapter overviews for the entire text, detailed remarks on each section, and comments on the exercise sets.

Printable Tests Printable tests are offered in TeX and Word format for every chapter, and can be customized by instructors.

PowerPoints Lecture Slides and PowerPoint Figures and Tables An extensive collection of PowerPoint slides for all chapters of the text are provided for instructor use. In addition, images of all figures and tables from the text are provided as PowerPoint slides.

Homework Delivery System An extensive homework delivery system, under development

for availability in fall 2012, will provide questions tied directly to the text, so that students will be able to do assignments on-line. Moreover, they will be able to use this system in a tutorial mode. This system will be able to automatically grade assignments, and deliver free form student input to instructors for their own analysis. Course management capabilities will be provided that will allow instructors to create assignments, automatically assign and grade homework, quiz, and test questions from a bank of questions tied directly to the text, create and edit their own questions, manage course announcements and due dates, and track student progress.

To the Student

What is discrete mathematics? Discrete mathematics is the part of mathematics devoted to the study of discrete objects. (Here *discrete* means consisting of distinct or unconnected elements.) The kinds of problems solved using discrete mathematics include:

- How many ways are there to choose a valid password on a computer system?
- What is the probability of winning a lottery?
- Is there a link between two computers in a network?
- How can I identify spam e-mail messages?
- How can I encrypt a message so that no unintended recipient can read it?
- What is the shortest path between two cities using a transportation system?
- How can a list of integers be sorted so that the integers are in increasing order?
- How many steps are required to do such a sorting?
- How can it be proved that a sorting algorithm correctly sorts a list?
- How can a circuit that adds two integers be designed?
- How many valid Internet addresses are there?

You will learn the discrete structures and techniques needed to solve problems such as these. More generally, discrete mathematics is used whenever objects are counted, when relationships between finite (or countable) sets are studied, and when processes involving a finite number of steps are analyzed. A key reason for the growth in the importance of discrete mathematics is that information is stored and manipulated by computing machines in a discrete fashion.

WHY STUDY DISCRETE MATHEMATICS? There are several important reasons for studying discrete mathematics. First, through this course you can develop your mathematical maturity: that is, your ability to understand and create mathematical arguments. You will not get very far in your studies in the mathematical sciences without these skills.

Second, discrete mathematics is the gateway to more advanced courses in all parts of the mathematical sciences. Discrete mathematics provides the mathematical foundations for many computer science courses including data structures, algorithms, database theory, automata theory, formal languages, compiler theory, computer security, and operating systems. Students find these courses much more difficult when they have not had the appropriate mathematical foundations from discrete math. One student has sent me an e-mail message saying that she used the contents of this book in every computer science course she took!

Math courses based on the material studied in discrete mathematics include logic, set theory, number theory, linear algebra, abstract algebra, combinatorics, graph theory, and probability theory (the discrete part of the subject).

Also, discrete mathematics contains the necessary mathematical background for solving problems in operations research (including many discrete optimization techniques), chemistry, engineering, biology, and so on. In the text, we will study applications to some of

these areas.

Many students find their introductory discrete mathematics course to be significantly more challenging than courses they have previously taken. One reason for this is that one of the primary goals of this course is to teach mathematical reasoning and problem solving, rather than a discrete set of skills. The exercises in this book are designed to reflect this goal. Although there are plenty of exercises in this text similar to those addressed in the examples, a large

percentage of the exercises require original thought. This is intentional. The material discussed in the text provides the tools needed to solve these exercises, but your job is to successfully apply these tools using your own creativity. One of the primary goals of this course is to learn how to attack problems that may be somewhat different from any you may have previously seen. Unfortunately, learning how to solve only particular types of exercises is not sufficient for success in developing the problem-solving skills needed in subsequent courses and professional work. This text addresses many different topics, but discrete mathematics is an extremely diverse and large area of study. One of my goals as an author is to help you develop the skills needed to master the additional material you will need in your own future pursuits.

THE EXERCISES I would like to offer some advice about how you can best learn discrete mathematics (and other subjects in the mathematical and computing sciences). You will learn the most by actively working exercises. I suggest that you solve as many as you possibly can. After working the exercises your instructor has assigned, I encourage you to solve additional exercises such as those in the exercise sets following each section of the text and in the supplementary exercises at the end of each chapter. (Note the key explaining the markings preceding exercises.)

Key to the Exercises

no marking A routine exercise

* A difficult exercise

** An extremely challenging exercise

An exercise containing a result used in the book (Table 1 on the following page shows where these exercises are used.)

(Requires calculus) An exercise whose solution requires the use of limits or concepts from differential or integral calculus

The best approach is to try exercises yourself before you consult the answer section at the end of this book. Note that the odd-numbered exercise answers provided in the text are answers only and not full solutions; in particular, the reasoning required to obtain answers is omitted in these answers. The *Student's Solutions Guide*, available separately, provides complete, worked solutions to all odd-numbered exercises in this text. When you hit an impasse trying to solve an odd-numbered exercise, I suggest you consult the *Student's Solutions Guide* and look for some guidance as to how to solve the problem. The more work you do yourself rather than passively reading or copying solutions, the more you will learn. The answers and solutions to the even numbered exercises are intentionally not available from the publisher; ask your instructor if you have trouble with these.

WEB RESOURCES You are *strongly* encouraged to take advantage of additional resources available on the Web, especially those on the companion website for this book found at www.mhhe.com/rosen. You will find many Extra Examples designed to clarify key concepts; Self Assessments for gauging how well you understand core topics; Interactive Demonstration Applets exploring key algorithms and other concepts; a Web Resources Guide containing an extensive selection of links to external sites relevant to the world of discrete mathematics; extra explanations and practice to help you master core concepts; added instruction on writing

proofs and on avoiding common mistakes in discrete mathematics; in-depth discussions of important applications; and guidance on utilizing MapleTM software to explore the computational aspects of discrete mathematics. Places in the text where these additional online resources are available are identified in the margins by special icons. You will also find (after fall 2012) the *Virtual Discrete Mathematics Tutor*, an on-line resource that provides extra support to help you make the transition from lower level courses to discrete mathematics. This tutorial should help answer many of your questions and correct errors that you may make, based on errors other students using this book, have made. For more details on these and other online resources, see the description of the companion website immediately preceding this “To the Student” message.

xx To the Student

TABLE 1 Hand-Icon Exercises and Where They Are Used			
<i>Section</i>	<i>Exercise</i>	<i>Section Where Used</i>	<i>Pages Where Used</i>
1.1	40	1.3	31
1.1	41	1.3	31
1.3	9	1.6	71
1.3	10	1.6	70, 71
1.3	15	1.6	71
1.3	30	1.6	71, 74
1.3	42	12.2	820
1.7	16	1.7	86
2.3	72	2.3	144
2.3	79	2.5	170
2.5	15	2.5	174
2.5	16	2.5	173
3.1	43	3.1	197
3.2	72	11.2	761
4.2	36	4.2	270
4.3	37	4.1	239
4.4	2	4.6	301
4.4	44	7.2	464
6.4	17	7.2	466
6.4	21	7.4	480
7.2	15	7.2	466
9.1	26	9.4	598
10.4	59	11.1	747
11.1	15	11.1	750
11.1	30	11.1	755

11.1	48	11.2	762
12.1	12	12.3	825
A.2	4	8.3	531

THE VALUE OF THIS BOOK My intention is to make your substantial investment in this text an excellent value. The book, the associated ancillaries, and companion website have taken many years of effort to develop and refine. I am confident that most of you will find that the text and associated materials will help you master discrete mathematics, just as so many previous students have. Even though it is likely that you will not cover some chapters in your current course, you should find it helpful—as many other students have—to read the relevant sections of the book as you take additional courses. Most of you will return to this book as a useful tool throughout your future studies, especially for those of you who continue in computer science, mathematics, and engineering. I have designed this book to be a gateway for future studies and explorations, and to be comprehensive reference, and I wish you luck as you begin your journey.

CHAPTER

1

- 1.1 Propositional Logic
- 1.2 Applications of Propositional Logic
- 1.3 Propositional Equivalences
- 1.4 Predicates and Quantifiers
- 1.5 Nested Quantifiers
- 1.6 Rules of Inference
- 1.7 Introduction to Proofs
- 1.8 Proof Methods and Strategy

The Foundations: Logic and Proofs

The rules of logic specify the meaning of mathematical statements. For instance, these rules

Kenneth H. Rosen
help us understand and reason with statements such as “There exists an integer that is not the sum of two squares” and “For every positive integer n , the sum of the positive integers not exceeding n is $n(n + 1)/2$.” Logic is the basis of all mathematical reasoning, and of all automated reasoning. It has practical applications to the design of computing machines, to the specification of systems, to artificial intelligence, to computer programming, to programming languages, and to other areas of computer science, as well as to many other fields of study.

To understand mathematics, we must understand what makes up a correct mathematical argument, that is, a proof. Once we prove a mathematical statement is true, we call it a theorem. A collection of theorems on a topic organize what we know about this topic. To learn a mathematical topic, a person needs to actively construct mathematical arguments on this topic, and not just read exposition. Moreover, knowing the proof of a theorem often makes it possible to modify the result to fit new situations.

Everyone knows that proofs are important throughout mathematics, but many people find it surprising how important proofs are in computer science. In fact, proofs are used to verify that computer programs produce the correct output for all possible input values, to show that algorithms always produce the correct result, to establish the security of a system, and to create artificial intelligence. Furthermore, automated reasoning systems have been created to allow computers to construct their own proofs.

In this chapter, we will explain what makes up a correct mathematical argument and introduce tools to construct these arguments. We will develop an arsenal of different proof methods that will enable us to prove many different types of results. After introducing many different methods of proof, we will introduce several strategies for

constructing proofs. We will introduce the notion of a conjecture and explain the process of developing mathematics by studying conjectures.

Propositional Logic

Introduction

The rules of logic give precise meaning to mathematical statements. These rules are used to distinguish between valid and invalid mathematical arguments. Because a major goal of this book is to teach the reader how to understand and how to construct correct mathematical arguments, we begin our study of discrete mathematics with an introduction to logic.

Besides the importance of logic in understanding mathematical reasoning, logic has numerous applications to computer science. These rules are used in the design of computer circuits, the construction of computer programs, the verification of the correctness of programs, and in many other ways. Furthermore, software systems have been developed for constructing some, but not all, types of proofs automatically. We will discuss these applications of logic in this and later chapters.

1

2 1 / The Foundations: Logic and Proofs

Propositions

Our discussion begins with an introduction to the basic building blocks of logic—propositions. A **proposition** is a declarative sentence (that is, a sentence that declares a fact) that is either true or false, but not both.

EXAMPLE 1 All the following declarative sentences are propositions.

1. Washington, D.C., is the capital of the United States of America.
2. Toronto is the capital of Canada.
3. $1 + 1 = 2$.
4. $2 + 2 = 3$.

Propositions 1 and 3 are true, whereas 2 and 4 are false.

Some sentences that are not propositions are given in Example 2.

EXAMPLE 2 Consider the following sentences.

1. What time is it?
2. Read this carefully.
3. $x + 1 = 2$.
4. $x + y = z$.

Sentences 1 and 2 are not propositions because they are not declarative sentences. Sentences 3 and 4 are not propositions because they are neither true nor false. Note that each of sentences 3 and 4 can be turned into a proposition if we assign values to the variables. We will also discuss

other ways to turn sentences such as these into propositions in Section 1.4.

We use letters to denote **propositional variables** (or **statement variables**), that is, variables that represent propositions, just as letters are used to denote numerical variables. The

ARISTOTLE (384 b.c.e.–322 b.c.e.) Aristotle was born in Stagirus (Stagira) in northern Greece. His father was the personal physician of the King of Macedonia. Because his father died when Aristotle was young, Aristotle could not follow the custom of following his father's profession. Aristotle became an orphan at a young age when his mother also died. His guardian who raised him taught him poetry, rhetoric, and Greek. At the age of 17, his guardian sent him to Athens to further his education. Aristotle joined Plato's Academy, where for 20 years he attended Plato's lectures, later presenting his own lectures on rhetoric. When Plato died in 347 B.C.E., Aristotle was not chosen to succeed him because his views differed too much from those of Plato. Instead, Aristotle joined the court of King Hermeas where he remained for three years, and married the niece of the King. When the Persians defeated Hermeas, Aristotle moved to Mytilene and, at the invitation of King Philip

of Macedonia, he tutored Alexander, Philip's son, who later became Alexander the Great. Aristotle tutored Alexander for five years and after the death of King Philip, he returned to Athens and set up his own school, called the Lyceum. Aristotle's followers were called the peripatetics, which means "to walk about," because Aristotle often walked around as he discussed philosophical questions. Aristotle taught at the Lyceum for 13 years where he lectured to his advanced students in the morning and gave popular lectures to a broad audience in the evening. When Alexander the Great died in 323 B.C.E., a backlash against anything related to Alexander led to trumped-up charges of impiety against Aristotle. Aristotle fled to Chalcis to avoid prosecution. He only lived one year in Chalcis, dying of a stomach ailment in 322 B.C.E.

Aristotle wrote three types of works: those written for a popular audience, compilations of scientific facts, and systematic treatises. The systematic treatises included works on logic, philosophy, psychology, physics, and natural history. Aristotle's writings were preserved by a student and were hidden in a vault where a wealthy book collector discovered them about 200 years later. They were taken to Rome, where they were studied by scholars and issued in new editions, preserving them for posterity.

1.1 Propositional Logic 3

conventional letters used for propositional variables are p, q, r, s, \dots . The **truth value** of a proposition is true, denoted by T, if it is a true proposition, and the truth value of a proposition is false, denoted by F, if it is a false proposition.

The area of logic that deals with propositions is called the **propositional calculus** or **propositional logic**. It was first developed systematically by the Greek philosopher Aristotle more than 2300 years ago.

We now turn our attention to methods for producing new propositions from those that we already have. These methods were discussed by the English mathematician George Boole in 1854 in his book *The Laws of Thought*. Many mathematical statements are constructed by combining one or more propositions. New propositions, called **compound propositions**, are formed from existing propositions using logical operators.

DEFINITION 1 Let p be a proposition. The *negation of p* , denoted by $\neg p$ (also denoted by \bar{p}), is the statement

"It is not the case that p ."

The proposition $\neg p$ is read "not p ." The truth value of the negation of p , $\neg p$, is the opposite of the truth value of p .

EXAMPLE 3 Find the negation of the proposition

"Michael's PC runs Linux"

and express this in simple English.

Solution: The negation is

"It is not the case that Michael's PC runs Linux."

This negation can be more simply expressed as

“Michael’s PC does not run Linux.”

EXAMPLE 4 Find the negation of the proposition

“Vandana’s smartphone has at least 32GB of memory”

and express this in simple English.

Solution: The negation is

“It is not the case that Vandana’s smartphone has at least 32GB of memory.”

This negation can also be expressed as

“Vandana’s smartphone does not have at least 32GB of memory”

or even more simply as

“Vandana’s smartphone has less than 32GB of memory.”

41 / The Foundations: Logic and Proofs

TABLE 1 The Truth Table for the Negation of a Proposition.	
p	$\neg p$
T	F
F	T

Table 1 displays the **truth table** for the negation of a proposition p . This table has a row for each of the two possible truth values of a proposition p . Each row shows the truth value of $\neg p$ corresponding to the truth value of p for this row.

The negation of a proposition can also be considered the result of the operation of the **negation operator** on a proposition. The negation operator constructs a new proposition from a single existing proposition. We will now introduce the logical operators that are used to form new propositions from two or more existing propositions. These logical operators are also called **connectives**.

DEFINITION 2 Let p and q be propositions. The *conjunction* of p and q , denoted by $p \wedge q$, is the proposition “ p and q .” The conjunction $p \wedge q$ is true when both p and q are true and is false otherwise.

Table 2 displays the truth table of $p \wedge q$. This table has a row for each of the four possible combinations of truth values of p and q . The four rows correspond to the pairs of truth values TT, TF, FT, and FF, where the first truth value in the pair is the truth value of p and the second truth value is the truth value of q .

Note that in logic the word “but” sometimes is used instead of “and” in a conjunction. For example, the statement “The sun is shining, but it is raining” is another way of saying “The sun is shining and it is raining.” (In natural language, there is a subtle difference in meaning between “and” and “but”; we will not be concerned with this nuance here.)

EXAMPLE 5 Find the conjunction of the propositions p and q where p is the proposition “Rebecca’s PC has more than 16 GB free hard disk space” and q is the proposition “The processor in Rebecca’s PC runs faster than 1 GHz.”

Solution: The conjunction of these propositions, $p \wedge q$, is the proposition “Rebecca’s PC has more than 16 GB free hard disk space, and the processor in Rebecca’s PC runs faster than 1 GHz.” This conjunction can be expressed more simply as “Rebecca’s PC has more than 16 GB free hard disk space, and its processor runs faster than 1 GHz.” For this conjunction to be true, both conditions given must be true. It is false, when one or both of these conditions are false. .

DEFINITION 3 Let p and q be propositions. The *disjunction* of p and q , denoted by $p \vee q$, is the proposition “ p or q .” The disjunction $p \vee q$ is false when both p and q are false and is true otherwise.

Table 3 displays the truth table for $p \vee q$.

TABLE 2 The T for the Conjunction Propositions.	
$p \ q$	
T	T
T	F
F	T
F	F

TABLE 3 The Truth Table for the Disjunction of Two Propositions.	
$p \ q$	$p \vee q$
T T	T
T F	T
F T	T
F F	F

1.1 Propositional Logic 5

The use of the connective *or* in a disjunction corresponds to one of the two ways the word *or* is used in English, namely, as an **inclusive or**. A disjunction is true when at least one of the two propositions is true. For instance, the inclusive or is being used in the statement

“Students who have taken calculus or computer science can take this class.”

Here, we mean that students who have taken both calculus and computer science can take the class, as well as the students who have taken only one of the two subjects. On the other hand, we are using the **exclusive or** when we say

“Students who have taken calculus or computer science, but not both, can enroll in this class.”

Here, we mean that students who have taken both calculus and a computer science course cannot take the class. Only those who have taken exactly one of the two courses can take the class. Similarly, when a menu at a restaurant states, “Soup or salad comes with an entrée,” the restaurant almost always means that customers can have either soup or salad, but not both. Hence, this is an exclusive, rather than an inclusive, or.

EXAMPLE 6 What is the disjunction of the propositions p and q where p and q are the same propositions as in Example 5?

Solution: The disjunction of p and q , $p \vee q$, is the proposition

“Rebecca’s PC has at least 16 GB free hard disk space, or the processor in Rebecca’s PC runs faster than 1 GHz.”

This proposition is true when Rebecca’s PC has at least 16 GB free hard disk space, when the PC’s processor runs faster than 1 GHz, and when both conditions are true. It is false when

both of these conditions are false, that is, when Rebecca's PC has less than 16 GB free hard disk space and the processor in her PC runs at 1 GHz or slower. .

As was previously remarked, the use of the connective *or* in a disjunction corresponds to one of the two ways the word *or* is used in English, namely, in an inclusive way. Thus, a disjunction is true when at least one of the two propositions in it is true. Sometimes, we use *or* in an exclusive sense. When the exclusive or is used to connect the propositions p and q , the proposition " p or q (but not both)" is obtained. This proposition is true when p is true and q is false, and when p is false and q is true. It is false when both p and q are false and when both are true.

GEORGE BOOLE (1815–1864) George Boole, the son of a cobbler, was born in Lincoln, England, in November 1815. Because of his family's difficult financial situation, Boole struggled to educate himself while supporting his family. Nevertheless, he became one of the most important mathematicians of the 1800s. Although he considered a career as a clergyman, he decided instead to go into teaching, and soon afterward opened a school of his own. In his preparation for teaching mathematics, Boole—unsatisfied with textbooks of his day—decided to read the works of the great mathematicians. While reading papers of the great French mathematician Lagrange, Boole made discoveries in the calculus of variations, the branch of analysis dealing with finding curves and surfaces by optimizing certain parameters.

In 1848 Boole published *The Mathematical Analysis of Logic*, the first of his contributions to symbolic logic. In 1849 he was appointed professor of mathematics at Queen's College in Cork, Ireland. In 1854 he published *The Laws of Thought*, his most famous work. In this book, Boole introduced what is now called *Boolean algebra* in his honor. Boole wrote textbooks on differential equations and on difference equations that were used in Great Britain until the end of the nineteenth century. Boole married in 1855; his wife was the niece of the professor of Greek at Queen's College. In 1864 Boole died from pneumonia, which he contracted as a result of keeping a lecture engagement even though he was soaking wet from a rainstorm.

61 / The Foundations: Logic and Proofs

TABLE 4 The Truth Table for the Exclusive Or of Two Propositions.

p q	$p \oplus q$
T T	F
T F	T
F T	T
F F	F

TABLE 5 The Truth Table for the Conditional Statement $p \rightarrow q$.

p q	$p \rightarrow q$
T T	T
T F	F
F T	T
F F	T

DEFINITION 4 Let p and q be propositions. The *exclusive or* of p and q , denoted by $p \oplus q$, is the proposition that is true when exactly one of p and q is true and is false otherwise.

The truth table for the exclusive or of two propositions is displayed in Table 4.

Conditional Statements

We will discuss several other important ways in which propositions can be combined.

DEFINITION 5 Let p and q be propositions. The *conditional statement* $p \rightarrow q$ is the proposition "if p , then q ." The conditional statement $p \rightarrow q$ is false when p is true and q is false, and true otherwise. In the conditional statement $p \rightarrow q$, p is called the *hypothesis* (or *antecedent* or *premise*) and q is called the *conclusion* (or *consequence*).

The statement $p \rightarrow q$ is called a conditional statement because $p \rightarrow q$ asserts that q is true on the condition that p holds. A conditional statement is also called an **implication**.

The truth table for the conditional statement $p \rightarrow q$ is shown in Table 5. Note that the statement $p \rightarrow q$ is true when both p and q are true and when p is false (no matter what truth value q has).

Because conditional statements play such an essential role in mathematical reasoning, a variety of terminology is used to express $p \rightarrow q$. You will encounter most if not all of the following ways to express this conditional statement:

“if p , then q ” “ p implies q ”
 “if p , q ” “ p only if q ”
 “ p is sufficient for q ” “a sufficient condition for q is p ”
 “ q if p ” “ q whenever p ”
 “ q when p ” “ q is necessary for p ”
 “a necessary condition for p is q ” “ q follows from p ”
 “ q unless $\neg p$ ”

A useful way to understand the truth value of a conditional statement is to think of an obligation or a contract. For example, the pledge many politicians make when running for office is

“If I am elected, then I will lower taxes.”

1.1 Propositional Logic 7

If the politician is elected, voters would expect this politician to lower taxes. Furthermore, if the politician is not elected, then voters will not have any expectation that this person will lower taxes, although the person may have sufficient influence to cause those in power to lower taxes. It is only when the politician is elected but does not lower taxes that voters can say that the politician has broken the campaign pledge. This last scenario corresponds to the case when p is true but q is false in $p \rightarrow q$.

Similarly, consider a statement that a professor might make:

“If you get 100% on the final, then you will get an A.”

If you manage to get a 100% on the final, then you would expect to receive an A. If you do not get 100% you may or may not receive an A depending on other factors. However, if you do get 100%, but the professor does not give you an A, you will feel cheated.

Of the various ways to express the conditional statement $p \rightarrow q$, the two that seem to cause the most confusion are “ p only if q ” and “ q unless $\neg p$.” Consequently, we will provide some guidance for clearing up this confusion.

To remember that “ p only if q ” expresses the same thing as “if p , then q ,” note that “ p only if q ” says that p cannot be true when q is not true. That is, the statement is false if p is true, but q is false. When p is false, q may be either true or false, because the statement says nothing about the truth value of q . Be careful not to use “ q only if p ” to express $p \rightarrow q$ because this is incorrect. To see this, note that the true values of “ q only if p ” and $p \rightarrow q$ are different when p and q have different truth values. [You might have trouble understanding how “unless” is used in conditional statements unless you read this paragraph carefully.](#) false, but it is true otherwise. Consequently, “ q unless $\neg p$ ” and $p \rightarrow q$ always have the same truth value. We illustrate the translation between conditional statements and English statements in Example 7.

To remember that “ q unless $\neg p$ ” expresses the same conditional statement as “if p , then q ,” note that “ q unless $\neg p$ ” means that if $\neg p$ is false, then q must be true. That is, the statement “ q unless $\neg p$ ” is false when p is true but q is

EXAMPLE 7 Let p be the statement “Maria learns discrete mathematics” and q the statement “Maria will find a good job.” Express the statement $p \rightarrow q$ as a statement in English.

Solution: From the definition of conditional statements, we see that when p is the statement “Maria learns discrete mathematics” and q is the statement “Maria will find a good job,” $p \rightarrow q$ represents the statement

“If Maria learns discrete mathematics, then she will find a good job.”

There are many other ways to express this conditional statement in English. Among the most natural of these are:

“Maria will find a good job when she learns discrete mathematics.”

“For Maria to get a good job, it is sufficient for her to learn discrete mathematics.”

and

“Maria will find a good job unless she does not learn discrete mathematics.”

Note that the way we have defined conditional statements is more general than the meaning attached to such statements in the English language. For instance, the conditional statement in Example 7 and the statement

“If it is sunny, then we will go to the beach.”

are statements used in normal language where there is a relationship between the hypothesis and the conclusion. Further, the first of these statements is true unless Maria learns discrete mathematics, but she does not get a good job, and the second is true unless it is indeed sunny, but we do not go to the beach. On the other hand, the statement

8 1 / The Foundations: Logic and Proofs

“If Juan has a smartphone, then $2 + 3 = 5$ ”

is true from the definition of a conditional statement, because its conclusion is true. (The truth value of the hypothesis does not matter then.) The conditional statement

“If Juan has a smartphone, then $2 + 3 = 6$ ”

is true if Juan does not have a smartphone, even though $2 + 3 = 6$ is false. We would not use these last two conditional statements in natural language (except perhaps in sarcasm), because there is no relationship between the hypothesis and the conclusion in either statement. In mathematical reasoning, we consider conditional statements of a more general sort than we use in English. The mathematical concept of a conditional statement is independent of a cause-and-effect relationship between hypothesis and conclusion. Our definition of a conditional statement specifies its truth values; it is not based on English usage. Propositional language is an artificial language; we only parallel English usage to make it easy to use and remember.

The if-then construction used in many programming languages is different from that used in logic. Most programming languages contain statements such as **if p then S** , where p is a proposition and S is a program segment (one or more statements to be executed). When execution of a program encounters such a statement, S is executed if p is true, but S is not executed if p is false, as illustrated in Example 8.

EXAMPLE 8 What is the value of the variable x after the statement

if $2 + 2 = 4$ then $x := x + 1$

if $x = 0$ before this statement is encountered? (The symbol $:=$ stands for assignment. The statement $x := x + 1$ means the assignment of the value of $x + 1$ to x .)

Solution: Because $2 + 2 = 4$ is true, the assignment statement $x := x + 1$ is executed. Hence, x has the value $0 + 1 = 1$ after this statement is encountered.

CONVERSE, CONTRAPOSITIVE, AND INVERSE We can form some new conditional statements starting with a conditional statement $p \rightarrow q$. In particular, there are three related

conditional statements that occur so often that they have special names. The proposition $q \rightarrow p$ is called the **converse** of $p \rightarrow q$. The **contrapositive** of $p \rightarrow q$ is the proposition $\neg q \rightarrow \neg p$. The proposition $\neg p \rightarrow \neg q$ is called the **inverse** of $p \rightarrow q$. We will see that of these three conditional statements formed from $p \rightarrow q$, only the contrapositive always has the same truth value as $p \rightarrow q$.

We first show that the contrapositive, $\neg q \rightarrow \neg p$, of a conditional statement $p \rightarrow q$ always has the same truth value as $p \rightarrow q$. To see this, note that the contrapositive is false only when $\neg p$ is false and $\neg q$ is true, that is, only when p is true and q is false. We now show that neither the converse, $q \rightarrow p$, nor the inverse, $\neg p \rightarrow \neg q$, has the same truth value as $p \rightarrow q$ for all possible truth values of p and q . Note that when p is true and q is false, the original conditional statement is false, but the converse and the inverse are both true. [Remember that the contrapositive, but neither the converse or inverse, of a conditional statement is equivalent to it.](#) study equivalent propositions in Section 1.3.) Take note that one of the most common logical errors is to assume that the converse or the inverse of a conditional statement is equivalent to this conditional statement.

When two compound propositions always have the same truth value we call them **equivalent**, so that a conditional statement and its contrapositive are equivalent. The converse and the inverse of a conditional statement are also equivalent, as the reader can verify, but neither is equivalent to the original conditional statement. (We will

We illustrate the use of conditional statements in Example 9.

1.1 Propositional Logic 9

EXAMPLE 9 What are the contrapositive, the converse, and the inverse of the conditional statement

“The home team wins whenever it is raining?”

Solution: Because “ q whenever p ” is one of the ways to express the conditional statement $p \rightarrow q$, the original statement can be rewritten as

“If it is raining, then the home team wins.”

Consequently, the contrapositive of this conditional statement is

“If the home team does not win, then it is not raining.”

The converse is

“If the home team wins, then it is raining.”

The inverse is

“If it is not raining, then the home team does not win.”

Only the contrapositive is equivalent to the original statement.

BICONDITIONALS We now introduce another way to combine propositions that expresses that two propositions have the same truth value.

DEFINITION 6 Let p and q be propositions. The *biconditional statement* $p \leftrightarrow q$ is the proposition “ p if and only if q .” The biconditional statement $p \leftrightarrow q$ is true when p and q have the same truth values, and is false otherwise. Biconditional statements are also called *bi-implications*.

The truth table for $p \leftrightarrow q$ is shown in Table 6. Note that the statement $p \leftrightarrow q$ is true when both the conditional statements $p \rightarrow q$ and $q \rightarrow p$ are true and is false otherwise. That is why we use the words “if and only if” to express this logical connective and why it is symbolically written by combining the symbols \rightarrow and \leftarrow . There are some other common ways to express $p \leftrightarrow q$:

“ p is necessary and sufficient for q ”

“if p then q , and conversely”
 “ p iff q .”

The last way of expressing the biconditional statement $p \leftrightarrow q$ uses the abbreviation “iff” for “if and only if.” Note that $p \leftrightarrow q$ has exactly the same truth value as $(p \rightarrow q) \wedge (q \rightarrow p)$.

TABLE 6 The Truth Table for the Biconditional $p \leftrightarrow q$.	
p	q
T	T
T	F
F	T
F	F

EXAMPLE 10 Let p be the statement “You can take the flight,” and let q be the statement “You buy a ticket.” Then $p \leftrightarrow q$ is the statement

“You can take the flight if and only if you buy a ticket.”

This statement is true if p and q are either both true or both false, that is, if you buy a ticket and can take the flight or if you do not buy a ticket and you cannot take the flight. It is false when p and q have opposite truth values, that is, when you do not buy a ticket, but you can take the flight (such as when you get a free trip) and when you buy a ticket but you cannot take the flight (such as when the airline bumps you).

IMPLICIT USE OF BICONDITIONALS You should be aware that biconditionals are not always explicit in natural language. In particular, the “if and only if” construction used in biconditionals is rarely used in common language. Instead, biconditionals are often expressed using an “if, then” or an “only if” construction. The other part of the “if and only if” is implicit. That is, the converse is implied, but not stated. For example, consider the statement in English “If you finish your meal, then you can have dessert.” What is really meant is “You can have dessert if and only if you finish your meal.” This last statement is logically equivalent to the two statements “If you finish your meal, then you can have dessert” and “You can have dessert only if you finish your meal.” Because of this imprecision in natural language, we need to make an assumption whether a conditional statement in natural language implicitly includes its converse. Because precision is essential in mathematics and in logic, we will always distinguish between the conditional statement $p \rightarrow q$ and the biconditional statement $p \leftrightarrow q$.

Truth Tables of Compound Propositions

We have now introduced four important logical connectives—conjunctions, disjunctions, conditional statements, and biconditional statements—as well as negations. We can use these connectives to build up complicated compound propositions involving any number of propositional variables. We can use truth tables to determine the truth values of these compound propositions, as Example 11 illustrates. We use a separate column to find the truth value of each compound expression that occurs in the compound proposition as it is built up. The truth values of the compound proposition for each combination of truth values of the propositional variables in it is found in the final column of the table.

EXAMPLE 11 Construct the truth table of the compound proposition

$$(p \vee \neg q) \rightarrow (p \wedge q).$$

Solution: Because this truth table involves two propositional variables p and q , there are four

rows in this truth table, one for each of the pairs of truth values TT, TF, FT, and FF. The first two columns are used for the truth values of p and q , respectively. In the third column we find the truth value of $\neg q$, needed to find the truth value of $p \vee \neg q$, found in the fourth column. The fifth column gives the truth value of $p \wedge q$. Finally, the truth value of $(p \vee \neg q) \rightarrow (p \wedge q)$ is found in the last column. The resulting truth table is shown in Table 7.

TABLE 7 The Truth Table of $(p \vee \neg q) \rightarrow (p \wedge q)$.				
$p \ q$	$\neg q$	$p \vee \neg q$	$p \wedge q$	$(p \vee \neg q) \rightarrow (p \wedge q)$
T T	F	T	T	T
T F	T	T	F	F
F T	F	F	F	T
F F	T	T	F	F

TABLE 8 Precedence of Logical Operators.	
Operator	Precedence
\neg	1
\wedge \vee	2 3
\rightarrow \leftrightarrow	4 5

Truth Value	Bit
T	1
F	0



1.1 Propositional Logic 11

Precedence of Logical Operators

We can construct compound propositions using the negation operator and the logical operators defined so far. We will generally use parentheses to specify the order in which logical operators in a compound proposition are to be applied. For instance, $(p \vee q) \wedge (\neg r)$ is the conjunction of $p \vee q$ and $\neg r$. However, to reduce the number of parentheses, we specify that the negation

operator is applied before all other logical operators. This means that $\neg p \wedge q$ is the conjunction of $\neg p$ and q , namely, $(\neg p) \wedge q$, not the negation of the conjunction of p and q , namely $\neg(p \wedge q)$.

Another general rule of precedence is that the conjunction operator takes precedence over the disjunction operator, so that $p \wedge q \vee r$ means $(p \wedge q) \vee r$ rather than $p \wedge (q \vee r)$. Because this rule may be difficult to remember, we will continue to use parentheses so that the order of the disjunction and conjunction operators is clear.

Finally, it is an accepted rule that the conditional and biconditional operators \rightarrow and \leftrightarrow have lower precedence than the conjunction and disjunction operators, \wedge and \vee . Consequently, $p \vee q \rightarrow r$ is the same as $(p \vee q) \rightarrow r$. We will use parentheses when the order of the conditional operator and biconditional operator is at issue, although the conditional operator has precedence over the biconditional operator. Table 8 displays the precedence levels of the logical operators, \neg , \wedge , \vee , \rightarrow , and \leftrightarrow .

Logic and Bit Operations

Computers represent information using bits. A **bit** is a symbol with two possible values, namely, 0 (zero) and 1 (one). This meaning of the word bit comes from *binary digit*, because zeros and ones are the digits used in binary representations of numbers. The well-known statistician John Tukey introduced this terminology in 1946. A bit can be used to represent a truth value, because there are two truth values, namely, *true* and *false*. As is customarily done, we will use a 1 bit to represent true and a 0 bit to represent false. That is, 1 represents T (true), 0 represents F (false). A variable is called a **Boolean variable** if its value is either true or false. Consequently, a Boolean variable can be represented using a bit.

Computer **bit operations** correspond to the logical connectives. By replacing true by a one and false by a zero in the truth tables for the operators \wedge , \vee , and \oplus , the tables shown in Table 9 for the corresponding bit operations are obtained. We will also use the notation *OR*, *AND*, and *XOR* for the operators \vee , \wedge , and \oplus , as is done in various programming languages.

JOHN WILDER TUKEY (1915–2000) Tukey, born in New Bedford, Massachusetts, was an only child. His parents, both teachers, decided home schooling would best develop his potential. His formal education began at Brown University, where he studied mathematics and chemistry. He received a master's degree in chemistry from Brown and continued his studies at Princeton University, changing his field of study from chemistry to

mathematics. He received his Ph.D. from Princeton in 1939 for work in topology, when he was appointed an instructor in mathematics at Princeton. With the start of World War II, he joined the Fire Control Research Office, where he began working in statistics. Tukey found statistical research to his liking and impressed several leading statisticians with his skills. In 1945, at the conclusion of the war, Tukey returned to the mathematics department at Princeton as a professor of statistics, and he also took a position at AT&T Bell Laboratories. Tukey founded

the Statistics Department at Princeton in 1966 and was its first chairman. Tukey made significant contributions to many areas of statistics, including the analysis of variance, the estimation of spectra of time series, inferences about the values of a set of parameters from a single experiment, and the philosophy of statistics. However, he is best known for his invention, with J. W. Cooley, of the fast Fourier transform. In addition to his contributions to statistics, Tukey was noted as a skilled wordsmith; he is credited with coining the terms *bit* and *software*.

Tukey contributed his insight and expertise by serving on the President's Science Advisory Committee. He chaired several important committees dealing with the environment, education, and chemicals and health. He also served on committees working on nuclear disarmament. Tukey received many awards, including the National Medal of Science.

HISTORICAL NOTE There were several other suggested words for a binary digit, including *binit* and *bigit*, that never were widely accepted. The adoption of the word *bit* may be due to its meaning as a common English word. For an account of Tukey's coining of the word *bit*, see the April 1984 issue of *Annals of the History of Computing*.

12 1 / The Foundations: Logic and Proofs

TABLE 9 Table for the Bit Operators OR, AND, and XOR.				
x	y	$x \vee y$	$x \wedge y$	$x \oplus y$
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	0

Information is often represented using bit strings, which are lists of zeros and ones. When this is done, operations on the bit strings can be used to manipulate this information.

DEFINITION 7 A *bit string* is a sequence of zero or more bits. The *length* of this string is the number of bits in the string.

EXAMPLE 12 101010011 is a bit string of length nine. .

We can extend bit operations to bit strings. We define the **bitwise OR**, **bitwise AND**, and **bitwise XOR** of two strings of the same length to be the strings that have as their bits the *OR*, *AND*, and *XOR* of the corresponding bits in the two strings, respectively. We use the symbols \vee , \wedge , and \oplus to represent the bitwise *OR*, bitwise *AND*, and bitwise *XOR* operations, respectively. We illustrate bitwise operations on bit strings with Example 13.

EXAMPLE 13 Find the bitwise *OR*, bitwise *AND*, and bitwise *XOR* of the bit strings 01 1011 0110 and 11 0001 1101. (Here, and throughout this book, bit strings will be split into blocks of four bits to make them easier to read.)

Solution: The bitwise *OR*, bitwise *AND*, and bitwise *XOR* of these strings are obtained by taking the *OR*, *AND*, and *XOR* of the corresponding bits, respectively. This gives us

01 1011 0110
 11 0001 1101
 11 1011 1111 **bitwise OR**
 01 0001 0100 **bitwise AND**

Exercises

- Which of these sentences are propositions? What are the truth values of those that are propositions?
 - Boston is the capital of Massachusetts.
 - Miami is the capital of Florida.
 - $2 + 3 = 5$.
 - $5 + 7 = 10$.
 - $x + 2 = 11$.
 - Answer this question.
- Which of these are propositions? What are the truth values of those that are propositions?
 - Do not pass go.
 - What time is it?
 - There are no black flies in Maine.
 - $4 + x = 5$.
 - The moon is made of green cheese.
 - $2^n \geq 100$.
- What is the negation of each of these propositions?
 - Mei has an MP3 player.
 - There is no pollution in New Jersey.
 - $2 + 1 = 3$.
 - The summer in Maine is hot and sunny.
- What is the negation of each of these propositions?
 - Jennifer and Teja are friends.
 - There are 13 items in a baker's dozen.
 - Abby sent more than 100 text messages every day.
 - 121 is a perfect square.
- What is the negation of each of these propositions?
 - Steve has more than 100 GB free disk space on his laptop.
 - Zach blocks e-mails and texts from Jennifer.
 - $7 \cdot 11 \cdot 13 = 999$.
 - Diane rode her bicycle 100 miles on Sunday.
- Suppose that Smartphone A has 256 MB RAM and 32 GB ROM, and the resolution of its camera is 8 MP; Smart phone B has 288 MB RAM and 64 GB ROM, and the resolution of its camera is 4 MP; and Smartphone C has 128 MB RAM and 32 GB ROM, and the resolution of its camera is 5 MP. Determine the truth value of each of these propositions.
 - Smartphone B has the most RAM of these three smart phones.
 - Smartphone C has more ROM or a higher resolution camera than Smartphone B.
 - Smartphone B has more RAM, more ROM, and a higher resolution camera than Smartphone A.
 - If Smartphone B has more RAM and more ROM than Smartphone C, then it also has a higher resolution camera.
 - Smartphone A has more RAM than Smartphone B if and only if Smartphone B has more RAM than Smart phone A.
- Suppose that during the most recent fiscal year, the annual revenue of Acme Computer was 138 billion dollars and its net profit was 8 billion dollars, the

annual revenue of Nadir Software was 87 billion dollars and its net profit was 5 billion dollars, and the annual revenue of Quixote Media was 111 billion dollars and its net profit was 13 billion dollars. Determine the truth value of each of these propositions for the most recent fiscal year.

- Quixote Media had the largest annual revenue.
- Nadir Software had the lowest net profit and Acme Computer had the largest annual revenue.
- Acme Computer had the largest net profit or Quixote Media had the largest net profit.

If Quixote Media had the smallest net profit, then Acme Computer had the largest annual revenue. e) Nadir Software had the smallest net profit if and only if Acme Computer had the largest annual revenue. **8.** Let p and q be the propositions

p : I bought a lottery ticket this week.

q : I won the million dollar jackpot.

Express each of these propositions as an English sentence.

- $\neg p$
- $p \vee q$
- $p \rightarrow q$
- $p \wedge q$
- $p \leftrightarrow q$
- $\neg p \rightarrow \neg q$
- $\neg p \wedge \neg q$
- $\neg p \vee (p \wedge q)$

9. Let p and q be the propositions "Swimming at the New Jersey shore is allowed" and "Sharks have been spotted near the shore," respectively. Express each of these compound propositions as an English sentence.

- $\neg p$
- $p \wedge q$
- $\neg p \vee q$
- $p \rightarrow \neg q$
- $\neg q \rightarrow p$
- $\neg p \rightarrow \neg q$
- $p \leftrightarrow \neg q$
- $\neg p \wedge (p \vee \neg q)$

1.1 Propositional Logic 13

10. Let p and q be the propositions "The election is decided" and "The votes have been counted," respectively. Express each of these compound propositions as an English sentence.

- $\neg p$
- $p \vee q$
- $\neg p \wedge q$
- $q \rightarrow p$
- $\neg q \rightarrow \neg p$
- $\neg p \rightarrow \neg q$
- $p \leftrightarrow q$
- $\neg q \vee (\neg p \wedge q)$

be the propositions

p : It is below freezing.

q : It is snowing.

Write these propositions using p and q and logical connectives (including negations).

- It is below freezing and snowing.
- It is below freezing but not snowing.
- It is not below freezing and it is not snowing.
- It is either snowing or below freezing (or both).
- If it is below freezing, it is also snowing.
- Either it is below freezing or it is snowing, but it is not snowing if it is below freezing.
- That it is below freezing is necessary and sufficient for it to be snowing.

12. Let p , q , and r be the propositions

p : You have the flu.

q : You miss the final examination.

r : You pass the course.

Express each of these propositions as an English sentence.

tence.

- a) $p \rightarrow q$ b) $\neg q \leftrightarrow r$
- c) $q \rightarrow \neg r$ d) $p \vee q \vee r$
- e) $(p \rightarrow \neg r) \vee (q \rightarrow \neg r)$
- f) $(p \wedge q) \vee (\neg q \wedge r)$

13. Let p and q be the propositions

p : You drive over 65 miles per hour.

q : You get a speeding ticket.

Write these propositions using p and q and logical connectives (including negations).

- a) You do not drive over 65 miles per hour.
- b) You drive over 65 miles per hour, but you do not get a speeding ticket.
- c) You will get a speeding ticket if you drive over 65 miles per hour.
- d) If you do not drive over 65 miles per hour, then you will not get a speeding ticket.
- e) Driving over 65 miles per hour is sufficient for getting a speeding ticket.
- f) You get a speeding ticket, but you do not drive over 65 miles per hour.
- g) Whenever you get a speeding ticket, you are driving over 65 miles per hour.

14. Let p , q , and r be the propositions

p : You get an A on the final exam.

q : You do every exercise in this book.

r : You get an A in this class.

Write these propositions using p , q , and r and logical connectives (including negations).

14 1 / The Foundations: Logic and Proofs

- a) You get an A in this class, but you do not do every exercise in this book.
- b) You get an A on the final, you do every exercise in this book, and you get an A in this class.
- c) To get an A in this class, it is necessary for you to get an A on the final.
- d) You get an A on the final, but you don't do every exercise in this book; nevertheless, you get an A in this class.
- e) Getting an A on the final and doing every exercise in this book is sufficient for getting an A in this class.
- f) You will get an A in this class if and only if you either do every exercise in this book or you get an A on the final.

15. Let p , q , and r be the propositions

p : Grizzly bears have been seen in the area.

q : Hiking is safe on the trail.

r : Berries are ripe along the trail.

Write these propositions using p , q , and r and logical connectives (including negations).

- a) Berries are ripe along the trail, but grizzly bears have not been seen in the area.
- b) Grizzly bears have not been seen in the area and hiking on the trail is safe, but berries are ripe along the trail.
- c) If berries are ripe along the trail, hiking is safe if and only if grizzly bears have not been seen in the area.
- d) It is not safe to hike on the trail, but grizzly bears have not been seen in the area and the berries along the trail are ripe.
- e) For hiking on the trail to be safe, it is necessary but

not sufficient that berries not be ripe along the trail and for grizzly bears not to have been seen in the area.

- f) Hiking is not safe on the trail whenever grizzly bears have been seen in the area and berries are ripe along the trail.

16. Determine whether these biconditionals are true or false.

- a) $2 + 2 = 4$ if and only if $1 + 1 = 2$.
- b) $1 + 1 = 2$ if and only if $2 + 3 = 4$.
- c) $1 + 1 = 3$ if and only if monkeys can fly.
- d) $0 > 1$ if and only if $2 > 1$.

17. Determine whether each of these conditional statements is true or false.

- a) If $1 + 1 = 2$, then $2 + 2 = 5$.
- b) If $1 + 1 = 3$, then $2 + 2 = 4$.
- c) If $1 + 1 = 3$, then $2 + 2 = 5$.
- d) If monkeys can fly, then $1 + 1 = 3$.

18. Determine whether each of these conditional statements is true or false.

- a) If $1 + 1 = 3$, then unicorns exist.
- b) If $1 + 1 = 3$, then dogs can fly.
- c) If $1 + 1 = 2$, then dogs can fly.
- d) If $2 + 2 = 4$, then $1 + 2 = 3$.

19. For each of these sentences, determine whether an inclusive or, or an exclusive or, is intended. Explain your answer.

- a) Coffee or tea comes with dinner.
- b) A password must have at least three digits or be at least eight characters long.
- c) The prerequisite for the course is a course in number theory or a course in cryptography.
- d) You can pay using U.S. dollars or euros.

20. For each of these sentences, determine whether an inclusive or, or an exclusive or, is intended. Explain your answer.

- a) Experience with C++ or Java is required.
- b) Lunch includes soup or salad.
- c) To enter the country you need a passport or a voter registration card.
- d) Publish or perish.

21. For each of these sentences, state what the sentence means if the logical connective or is an inclusive or (that is, a disjunction) versus an exclusive or. Which of these meanings or do you think is intended?

- a) To take discrete mathematics, you must have taken calculus or a course in computer science.
- b) When you buy a new car from Acme Motor Company, you get \$2000 back in cash or a 2% car loan.

c) Dinner for two includes two items from column A or three items from column B.

d) School is closed if more than 2 feet of snow falls or if the wind chill is below -100.

22. Write each of these statements in the form "if p , then q " in English. [Hint: Refer to the list of common ways to express conditional statements provided in this section.] a) It is necessary to wash the boss's car to get promoted. b) Winds from the south imply a

spring thaw. **c)** A sufficient condition for the warranty to be good is

that you bought the computer less than a year ago. **d)** Willy gets caught whenever he cheats.

e) You can access the website only if you pay a subscription fee.

f) Getting elected follows from knowing the right people.

g) Carol gets seasick whenever she is on a boat. **23.**

Write each of these statements in the form “if p , then q ” in English. [Hint: Refer to the list of common ways to express conditional statements.]

a) It snows whenever the wind blows from the northeast. **b)** The apple trees will bloom if it stays warm for a week. **c)** That the Pistons win the championship implies that they beat the Lakers.

d) It is necessary to walk 8 miles to get to the top of Long’s Peak.

e) To get tenure as a professor, it is sufficient to be world famous.

f) If you drive more than 400 miles, you will need to buy gasoline.

g) Your guarantee is good only if you bought your CD player less than 90 days ago.

h) Jan will go swimming unless the water is too cold.

24. Write each of these statements in the form “if p , then q ” in English. [Hint: Refer to the list of common ways to express conditional statements provided in this section.]

a) I will remember to send you the address only if you send me an e-mail message.

b) To be a citizen of this country, it is sufficient that you were born in the United States.

c) If you keep your textbook, it will be a useful reference in your future courses.

d) The Red Wings will win the Stanley Cup if their goalie plays well.

e) That you get the job implies that you had the best credentials.

f) The beach erodes whenever there is a storm. **g)** It is necessary to have a valid password to log on to the server.

h) You will reach the summit unless you begin your climb too late.

25. Write each of these propositions in the form “ p if and only if q ” in English.

a) If it is hot outside you buy an ice cream cone, and if you buy an ice cream cone it is hot outside.

b) For you to win the contest it is necessary and sufficient that you have the only winning ticket.

c) You get promoted only if you have connections, and you have connections only if you get promoted.

d) If you watch television your mind will decay, and conversely.

e) The trains run late on exactly those days when I take it.

26. Write each of these propositions in the form “ p if and only if q ” in English.

a) For you to get an A in this course, it is necessary and sufficient that you learn how to solve discrete mathematics problems.

b) If you read the newspaper every day, you will be informed, and conversely.

c) It rains if it is a weekend day, and it is a weekend day if it rains.

d) You can see the wizard only if the wizard is not in, and the wizard is not in only if you can see him. **27.** State the converse, contrapositive, and inverse of each of these conditional statements.

a) If it snows today, I will ski tomorrow.

b) I come to class whenever there is going to be a quiz. **c)** A positive integer is a prime only if it has no divisors other than 1 and itself.

28. State the converse, contrapositive, and inverse of each of these conditional statements.

a) If it snows tonight, then I will stay at home. **b)** I go to the beach whenever it is a sunny summer day. **c)** When I stay up late, it is necessary that I sleep until noon.

29. How many rows appear in a truth table for each of these compound propositions?

a) $p \rightarrow \neg p$

b) $(p \vee \neg r) \wedge (q \vee \neg s)$

1.1 Propositional Logic 15

c) $q \vee p \vee \neg s \vee \neg r \vee \neg t \vee u$

d) $(p \wedge r \wedge t) \leftrightarrow (q \wedge t)$

30. How many rows appear in a truth table for each of these compound propositions?

a) $(q \rightarrow \neg p) \vee (\neg p \rightarrow \neg q)$

b) $(p \vee \neg t) \wedge (p \vee \neg s)$

c) $(p \rightarrow r) \vee (\neg s \rightarrow \neg t) \vee (\neg u \rightarrow v)$

d) $(p \wedge r \wedge s) \vee (q \wedge t) \vee (r \wedge \neg t)$

31. Construct a truth table for each of these compound propositions.

a) $p \wedge \neg p$ **b)** $p \vee \neg p$

c) $(p \vee \neg q) \rightarrow q$ **d)** $(p \vee q) \rightarrow (p \wedge q)$ **e)** $(p \rightarrow q) \leftrightarrow (\neg q \rightarrow \neg p)$

f) $(p \rightarrow q) \rightarrow (q \rightarrow p)$

32. Construct a truth table for each of these compound propositions.

a) $p \rightarrow \neg p$ **b)** $p \leftrightarrow \neg p$

c) $p \oplus (p \vee q)$ **d)** $(p \wedge q) \rightarrow (p \vee q)$ **e)** $(q \rightarrow \neg p) \leftrightarrow (p \leftrightarrow q)$

f) $(p \leftrightarrow q) \oplus (p \leftrightarrow \neg q)$

33. Construct a truth table for each of these compound propositions.

a) $(p \vee q) \rightarrow (p \oplus q)$ **b)** $(p \oplus q) \rightarrow (p \wedge q)$ **c)** $(p \vee q) \oplus (p \wedge q)$ **d)** $(p \leftrightarrow q) \oplus (\neg p \leftrightarrow q)$ **e)** $(p \leftrightarrow q) \oplus (\neg p \leftrightarrow \neg r)$

f) $(p \oplus q) \rightarrow (p \oplus \neg q)$

34. Construct a truth table for each of these compound propositions.

a) $p \oplus p$ **b)** $p \oplus \neg p$

c) $p \oplus \neg q$ **d)** $\neg p \oplus \neg q$

e) $(p \oplus q) \vee (p \oplus \neg q)$ **f)** $(p \oplus q) \wedge (p \oplus \neg q)$ **35.**

Construct a truth table for each of these compound propositions.

a) $p \rightarrow \neg q$ **b)** $\neg p \leftrightarrow q$

c) $(p \rightarrow q) \vee (\neg p \rightarrow q)$ **d)** $(p \rightarrow q) \wedge (\neg p \rightarrow q)$ **e)**

$$(p \leftrightarrow q) \vee (\neg p \leftrightarrow q)$$

$$f) (\neg p \leftrightarrow \neg q) \leftrightarrow (p \leftrightarrow q)$$

36. Construct a truth table for each of these compound propositions.

$$a) (p \vee q) \vee r \quad b) (p \vee q) \wedge r$$

$$c) (p \wedge q) \vee r \quad d) (p \wedge q) \wedge r$$

$$e) (p \vee q) \wedge \neg r \quad f) (p \wedge q) \vee \neg r$$

37. Construct a truth table for each of these compound propositions.

$$a) p \rightarrow (\neg q \vee r)$$

$$b) \neg p \rightarrow (q \rightarrow r)$$

$$c) (p \rightarrow q) \vee (\neg p \rightarrow r)$$

$$d) (p \rightarrow q) \wedge (\neg p \rightarrow r)$$

$$e) (p \leftrightarrow q) \vee (\neg q \leftrightarrow r)$$

$$f) (\neg p \leftrightarrow \neg q) \leftrightarrow (q \leftrightarrow r)$$

38. Construct a truth table for $((p \rightarrow q) \rightarrow r) \rightarrow$

s. 39. Construct a truth table for $(p \leftrightarrow q) \leftrightarrow (r \leftrightarrow s)$.

16 1 / The Foundations: Logic and Proofs

40. Explain, without using a truth table, why $(p \vee \neg q) \wedge (q \vee \neg r) \wedge (r \vee \neg p)$ is true when p , q , and r have the same truth value and it is false otherwise.

41. Explain, without using a truth table, why $(p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$ is true when at least one of p , q , and r is true and at least one is false, but is false when all three variables have the same truth value.

42. What is the value of x after each of these statements is encountered in a computer program, if $x = 1$ before the statement is reached?

a) if $x + 2 = 3$ then $x := x + 1$

b) if $(x + 1 = 3)$ OR $(2x + 2 = 3)$ then $x := x + 1$ c)

if $(2x + 3 = 5)$ AND $(3x + 4 = 7)$ then $x := x + 1$ d)

if $(x + 1 = 2)$ XOR $(x + 2 = 3)$ then $x := x + 1$ e) if $x < 2$ then $x := x + 1$

43. Find the bitwise OR, bitwise AND, and bitwise XOR of each of these pairs of bit strings.

a) 101 1110, 010 0001

b) 1111 0000, 1010 1010

c) 00 0111 0001, 10 0100 1000

d) 11 1111 1111, 00 0000 0000

44. Evaluate each of these expressions.

a) $1\ 1000 \wedge (0\ 1011 \vee 1\ 1011)$

b) $(0\ 1111 \wedge 1\ 0101) \vee 0\ 1000$

c) $(0\ 1010 \oplus 1\ 1011) \oplus 0\ 1000$

$$d) (1\ 1011 \vee 0\ 1010) \wedge (1\ 0001 \vee 1\ 1011)$$

Fuzzy logic is used in artificial intelligence. In fuzzy logic, a proposition has a truth value that is a number between 0 and 1, inclusive. A proposition with a truth value of 0 is false and one with a truth value of 1 is true. Truth values that are between 0 and 1 indicate varying degrees of truth. For instance, the truth value 0.8 can be assigned to the statement “Fred is happy,”

because Fred is happy most of the time, and the truth value 0.4 can be assigned to the statement “John is happy,” because John is happy slightly less than half the time. Use these truth values to solve Exercises 45–47.

45. The truth value of the negation of a proposition in fuzzy logic is 1 minus the truth value of the proposition. What are the truth values of the statements “Fred is not happy” and “John is not happy?”

46. The truth value of the conjunction of two propositions in fuzzy logic is the minimum of the truth values of the two propositions. What are the truth values of the statements “Fred and John are happy” and “Neither Fred nor John is happy?”

47. The truth value of the disjunction of two propositions in fuzzy logic is the maximum of the truth values of the two propositions. What are the truth values of the statements “Fred is happy, or John is happy” and “Fred is not happy, or John is not happy?”

* 48. Is the assertion “This statement is false” a

proposition? * 49. The n th statement in a list of 100 statements is “Exactly n of the statements in this list are false.”

a) What conclusions can you draw from these statements?

b) Answer part (a) if the n th statement is “At least n of the statements in this list are false.”

c) Answer part (b) assuming that the list contains 99 statements.

50. An ancient Sicilian legend says that the barber in a remote town who can be reached only by traveling a dangerous mountain road shaves those people, and only those people, who do not shave themselves. Can there be such a barber?

Applications of Propositional Logic

Introduction

Logic has many important applications to mathematics, computer science, and numerous other disciplines. Statements in mathematics and the sciences and in natural language often are imprecise or ambiguous. To make such statements precise, they can be translated into the language of logic. For example, logic is used in the specification of software and hardware, because these specifications need to be precise before development begins. Furthermore, propositional logic and its rules can be used to design computer circuits, to construct computer programs, to verify the correctness of programs, and to build expert systems. Logic

can be used to analyze and solve many familiar puzzles. Software systems based on the rules of logic have been developed for constructing some, but not all, types of proofs automatically. We will discuss some of these applications of propositional logic in this section and in later chapters.

Translating English Sentences

There are many reasons to translate English sentences into expressions involving propositional variables and logical connectives. In particular, English (and every other human language) is

1.2 Applications of Propositional Logic 17

often ambiguous. Translating sentences into compound statements (and other types of logical expressions, which we will introduce later in this chapter) removes the ambiguity. Note that this may involve making a set of reasonable assumptions based on the intended meaning of the sentence. Moreover, once we have translated sentences from English into logical expressions we can analyze these logical expressions to determine their truth values, we can manipulate them, and we can use rules of inference (which are discussed in Section 1.6) to reason about them.

To illustrate the process of translating an English sentence into a logical expression, consider Examples 1 and 2.

EXAMPLE 1 How can this English sentence be translated into a logical expression?

“You can access the Internet from campus only if you are a computer science major or you are not a freshman.”

Solution: There are many ways to translate this sentence into a logical expression. Although it is possible to represent the sentence by a single propositional variable, such as p , this would not be useful when analyzing its meaning or reasoning with it. Instead, we will use propositional variables to represent each sentence part and determine the appropriate logical connectives between them. In particular, we let a , c , and f represent “You can access the Internet from campus,” “You are a computer science major,” and “You are a freshman,” respectively. Noting that “only if” is one way a conditional statement can be expressed, this sentence can be represented as

$$a \rightarrow (c \vee \neg f).$$

EXAMPLE 2 How can this English sentence be translated into a logical expression?

“You cannot ride the roller coaster if you are under 4 feet tall unless you are older than 16 years old.”

Solution: Let q , r , and s represent “You can ride the roller coaster,” “You are under 4 feet tall,” and “You are older than 16 years old,” respectively. Then the sentence can be translated to

$$(r \wedge \neg s) \rightarrow \neg q.$$

Of course, there are other ways to represent the original sentence as a logical expression, but the one we have used should meet our needs.

System Specifications

Translating sentences in natural language (such as English) into logical expressions is an essential part of specifying both hardware and software systems. System and software engineers take requirements in natural language and produce precise and unambiguous specifications that can be used as the basis for system development. Example 3 shows how compound propositions can be used in this process.

EXAMPLE 3 Express the specification “The automated reply cannot be sent when the file system is full” using logical connectives.

Solution: One way to translate this is to let p denote “The automated reply can be sent” and q denote “The file system is full.” Then $\neg p$ represents “It is not the case that the automated

18 1 / The Foundations: Logic and Proofs

reply can be sent,” which can also be expressed as “The automated reply cannot be sent.” Consequently, our specification can be represented by the conditional statement $q \rightarrow \neg p$.

System specifications should be **consistent**, that is, they should not contain conflicting requirements that could be used to derive a contradiction. When specifications are not consistent, there would be no way to develop a system that satisfies all specifications.

EXAMPLE 4 Determine whether these system specifications are consistent:

“The diagnostic message is stored in the buffer or it is retransmitted.”

“The diagnostic message is not stored in the buffer.”

“If the diagnostic message is stored in the buffer, then it is retransmitted.”

Solution: To determine whether these specifications are consistent, we first express them using logical expressions. Let p denote “The diagnostic message is stored in the buffer” and let q denote “The diagnostic message is retransmitted.” The specifications can then be written as $p \vee q$, $\neg p$, and $p \rightarrow q$. An assignment of truth values that makes all three specifications true must have p false to make $\neg p$ true. Because we want $p \vee q$ to be true but p must be false, q must be true. Because $p \rightarrow q$ is true when p is false and q is true, we conclude that these specifications are consistent, because they are all true when p is false and q is true. We could come to the same conclusion by use of a truth table to examine the four possible assignments of truth values to p and q .

EXAMPLE 5 Do the system specifications in Example 4 remain consistent if the specification “The diagnostic message is not retransmitted” is added?

Solution: By the reasoning in Example 4, the three specifications from that example are true only in the case when p is false and q is true. However, this new specification is $\neg q$, which is false when q is true. Consequently, these four specifications are inconsistent.

Boolean Searches

Logical connectives are used extensively in searches of large collections of information, such as indexes of Web pages. Because these searches employ techniques from propositional logic, they are called **Boolean searches**.

In Boolean searches, the connective *AND* is used to match records that contain both of two search terms, the connective *OR* is used to match one or both of two search terms, and the connective *NOT* (sometimes written as *AND NOT*) is used to exclude a particular search term. Careful planning of how logical connectives are used is often required when Boolean searches are used to locate information of potential interest. Example 6 illustrates how Boolean searches are carried out.

EXAMPLE 6 Web Page Searching Most Web search engines support Boolean searching techniques, which usually can help find Web pages about particular subjects. For instance, using Boolean searching to find Web pages about universities in New Mexico, we can look for pages matching `NEW AND MEXICO AND UNIVERSITIES`. The results of this search will include those pages that contain the three words `NEW`, `MEXICO`, and `UNIVERSITIES`. This will include all of the pages of interest, together with others such as a page about new universities in Mexico. (Note that in Google, and many other search engines, the word “AND” is not needed, although it is understood, because all search terms are included by default. These search engines also support the use of quotation marks to search for specific phrases. So, it may be more effective to search for pages matching “New Mexico” `AND UNIVERSITIES`.)

1.2 Applications of Propositional Logic 19

Next, to find pages that deal with universities in New Mexico or Arizona, we can search for pages matching `(NEW AND MEXICO OR ARIZONA) AND UNIVERSITIES`. (*Note:* Here the `AND` operator takes precedence over the `OR` operator. Also, in Google, the terms used for this search would be `NEW MEXICO OR ARIZONA`.) The results of this search will include all pages that contain the word `UNIVERSITIES` and either both the words `NEW` and `MEXICO` or the word `ARIZONA`. Again, pages besides those of interest will be listed. Finally, to find Web pages that deal with universities in Mexico (and not New Mexico), we might first look for pages matching `MEXICO AND UNIVERSITIES`, but because the results of this search will include pages about universities in New Mexico, as well as universities in Mexico, it might be better to search for pages matching `(MEXICO AND UNIVERSITIES) NOT NEW`. The results of this search include pages that contain both the words `MEXICO` and `UNIVERSITIES` but do not contain the word `NEW`. (In Google, and many other search engines, the word “NOT” is replaced by the symbol “-”. In Google, the terms used for this last search would be `MEXICO UNIVERSITIES -NEW`.)

Logic Puzzles

Puzzles that can be solved using logical reasoning are known as **logic puzzles**. Solving logic puzzles is an excellent way to practice working with the rules of logic. Also, computer programs designed to carry out logical reasoning often use well-known logic puzzles to illustrate their capabilities. Many people enjoy solving logic puzzles, published in periodicals, books, and on the Web, as a recreational activity.

We will discuss two logic puzzles here. We begin with a puzzle originally posed by Raymond Smullyan, a master of logic puzzles, who has published more than a dozen books containing challenging puzzles that involve logical reasoning. In Section 1.3 we will also discuss the extremely popular logic puzzle Sudoku.

EXAMPLE 7 In [Sm78] Smullyan posed many puzzles about an island that has two kinds of inhabitants, knights, who always tell the truth, and their opposites, knaves, who always lie. You encounter two people *A* and *B*. What are *A* and *B* if *A* says “*B* is a knight” and *B* says “The two of us are opposite types?”

Solution: Let *p* and *q* be the statements that *A* is a knight and *B* is a knight, respectively, so that $\neg p$ and $\neg q$ are the statements that *A* is a knave and *B* is a knave, respectively.

We first consider the possibility that *A* is a knight; this is the statement that *p* is true. If *A* is a knight, then he is telling the truth when he says that *B* is a knight, so that *q* is true, and *A* and *B* are the same type. However, if *B* is a knight, then *B*’s statement that *A* and *B* are of opposite types, the statement $(p \wedge \neg q) \vee (\neg p \wedge q)$, would have to be true, which it is not, because *A* and *B* are both knights. Consequently, we can conclude that *A* is not a knight, that is, that *p* is false.

If *A* is a knave, then because everything a knave says is false, *A*’s statement that *B* is a knight, that is, that *q* is true, is a lie. This means that *q* is false and *B* is also a knave.

Furthermore, if B is a knave, then B 's statement that A and B are opposite types is a lie, which is consistent with both A and B being knaves. We can conclude that both A and B are knaves.

We pose more of Smullyan's puzzles about knights and knaves in Exercises 19–23. In Exercises 24–31 we introduce related puzzles where we have three types of people, knights and knaves as in this puzzle together with spies who can lie.

Next, we pose a puzzle known as the **muddy children puzzle** for the case of two children.

20 1 / The Foundations: Logic and Proofs

EXAMPLE 8 A father tells his two children, a boy and a girl, to play in their backyard without getting dirty. However, while playing, both children get mud on their foreheads. When the children stop playing, the father says “At least one of you has a muddy forehead,” and then asks the children to answer “Yes” or “No” to the question: “Do you know whether you have a muddy forehead?” The father asks this question twice. What will the children answer each time this question is asked, assuming that a child can see whether his or her sibling has a muddy forehead, but cannot see his or her own forehead? Assume that both children are honest and that the children answer each question simultaneously.

Solution: Let s be the statement that the son has a muddy forehead and let d be the statement that the daughter has a muddy forehead. When the father says that at least one of the two children has a muddy forehead, he is stating that the disjunction $s \vee d$ is true. Both children will answer “No” the first time the question is asked because each sees mud on the other child's forehead. That is, the son knows that d is true, but does not know whether s is true, and the daughter knows that s is true, but does not know whether d is true.

After the son has answered “No” to the first question, the daughter can determine that d must be true. This follows because when the first question is asked, the son knows that $s \vee d$ is true, but cannot determine whether s is true. Using this information, the daughter can conclude that d must be true, for if d were false, the son could have reasoned that because $s \vee d$ is true, then s must be true, and he would have answered “Yes” to the first question. The son can reason in a similar way to determine that s must be true. It follows that both children answer “Yes” the second time the question is asked.

Logic Circuits

Propositional logic can be applied to the design of computer hardware. This was first observed in 1938 by Claude Shannon in his MIT master's thesis. In Chapter 12 we will study this topic in depth. (See that chapter for a biography of Shannon.) We give a brief introduction to this application here.

A **logic circuit** (or **digital circuit**) receives input signals p_1, p_2, \dots, p_n , each a bit [either 0 (off) or 1 (on)], and produces output signals s_1, s_2, \dots, s_n , each a bit. In this section we will

In Chapter 12 we design some useful circuits.



restrict our attention to logic circuits with a single output signal; in general, digital circuits may have multiple outputs.

RAYMOND SMULLYAN (BORN 1919) Raymond Smullyan dropped out of high school. He wanted to study what he was really interested in and not standard high school material. After jumping from one university to the next, he earned an undergraduate degree in mathematics at the University of Chicago in 1955. He paid his college expenses by performing magic tricks at parties and clubs. He obtained a Ph.D. in logic in 1959 at Princeton, studying under Alonzo Church. After graduating from Princeton, he taught mathematics and logic at Dartmouth College, Princeton University, Yeshiva University, and the City University of New York. He joined the philosophy department at Indiana University in 1981 where he is now an emeritus professor. Smullyan has written many books on recreational logic and

mathematics, including *Satan, Cantor, and Infinity*; *What Is the Name of This Book?*; *The Lady or the Tiger?*; *Alice in Puzzleland*;

Forever Undecided; and *The Riddle of Scheherazade: Amazing Logic Puzzles, Ancient and Modern*. Because his logic puzzles are challenging, entertaining, and thought-provoking, he is considered to be a modern-day Lewis Carroll. Smullyan has also written several books about the application of deductive logic to chess, three collections of philosophical essays and aphorisms, and several advanced books on mathematical logic and set theory. He is particularly interested in self-reference and has worked on extending some of Gödel's results that show that it is impossible to write a computer program that can solve all mathematical problems. He is also particularly interested in explaining ideas from mathematical logic to the public.

Smullyan is a talented musician and often plays piano with his wife, who is a concert-level pianist. Making telescopes is one of his hobbies. He is also interested in optics and stereo photography. He states "I've never had a conflict between teaching and research as some people do because when I'm teaching, I'm doing research." Smullyan is the subject of a documentary short film entitled *This Film Needs No Title*.

1.2 Applications of Propositional Logic 21



FIGURE 1 Basic logic gates.

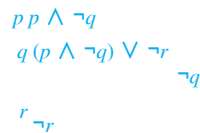


FIGURE 2 A combinatorial circuit.

Complicated digital circuits can be constructed from three basic circuits, called **gates**, shown in Figure 1. The **inverter**, or **NOT gate**, takes an input bit p , and produces as output $\neg p$. The **OR gate** takes two input signals p and q , each a bit, and produces as output the signal $p \vee q$. Finally, the **AND gate** takes two input signals p and q , each a bit, and produces as output the signal $p \wedge q$. We use combinations of these three basic gates to build more complicated circuits, such as that shown in Figure 2.

Given a circuit built from the basic logic gates and the inputs to the circuit, we determine the output by tracing through the circuit, as Example 9 shows.

EXAMPLE 9 Determine the output for the combinatorial circuit in Figure 2.

Solution: In Figure 2 we display the output of each logic gate in the circuit. We see that the AND gate takes input of p and $\neg q$, the output of the inverter with input q , and produces $p \wedge \neg q$. Next, we note that the OR gate takes input $p \wedge \neg q$ and $\neg r$, the output of the inverter with input r , and produces the final output $(p \wedge \neg q) \vee \neg r$.

Suppose that we have a formula for the output of a digital circuit in terms of negations, disjunctions, and conjunctions. Then, we can systematically build a digital circuit with the desired output, as illustrated in Example 10.

EXAMPLE 10 Build a digital circuit that produces the output $(p \vee \neg r) \wedge (\neg p \vee (q \vee \neg r))$ when given input bits p , q , and r .

Solution: To construct the desired circuit, we build separate circuits for $p \vee \neg r$ and for $\neg p \vee (q \vee \neg r)$ and combine them using an AND gate. To construct a circuit for $p \vee \neg r$, we use an inverter to produce $\neg r$ from the input r . Then, we use an OR gate to combine p and $\neg r$. To

build a circuit for $\neg p \vee (q \vee \neg r)$, we first use an inverter to obtain $\neg r$. Then we use an OR gate with inputs q and $\neg r$ to obtain $q \vee \neg r$. Finally, we use another inverter and an OR gate to get $\neg p \vee (q \vee \neg r)$ from the inputs p and $q \vee \neg r$.

To complete the construction, we employ a final AND gate, with inputs $p \vee \neg r$ and $\neg p \vee (q \vee \neg r)$. The resulting circuit is displayed in Figure 3.

We will study logic circuits in great detail in Chapter 12 in the context of Boolean algebra, and with different notation.

22 1 / The Foundations: Logic and Proofs

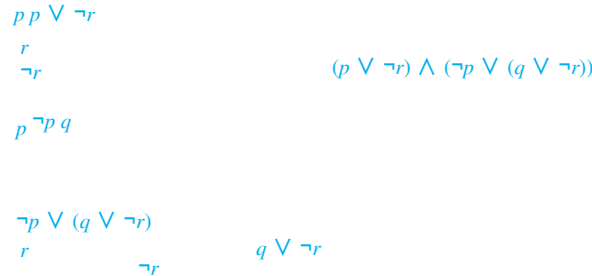


FIGURE 3 The circuit for $(p \vee \neg r) \wedge (\neg p \vee (q \vee \neg r))$.

Exercises

In Exercises 1–6, translate the given statement into propositional logic using the propositions provided.

1. You cannot edit a protected Wikipedia entry unless you are an administrator. Express your answer in terms of e : “You can edit a protected Wikipedia entry” and a : “You are an administrator.”
2. You can see the movie only if you are over 18 years old or you have the permission of a parent. Express your answer in terms of m : “You can see the movie,” e : “You are over 18 years old,” and p : “You have the permission of a parent.”
3. You can graduate only if you have completed the requirements of your major and you do not owe money to the university and you do not have an overdue library book. Express your answer in terms of g : “You can graduate,” m : “You owe money to the university,” r : “You have completed the requirements of your major,” and b : “You have an overdue library book.”
4. To use the wireless network in the airport you must pay the daily fee unless you are a subscriber to the service. Express your answer in terms of w : “You can use the wireless network in the airport,” d : “You pay the daily fee,” and s : “You are a subscriber to the service.”
5. You are eligible to be President of the U.S.A. only if you are at least 35 years old, were born in the U.S.A., or at the time of your birth both of your parents were citizens, and you have lived at least 14 years in the country. Express your answer in terms of e : “You are eligible to be President of the U.S.A.,” a : “You are at least 35 years old,” b : “You were born in the U.S.A.,” p : “At the time of your birth, both of your parents were

citizens,” and r : “You have lived at least 14 years in the U.S.A.”

6. You can upgrade your operating system only if you have a 32-bit processor running at 1 GHz or faster, at least 1 GB RAM, and 16 GB free hard disk space, or a 64-bit processor running at 2 GHz or faster, at least 2 GB RAM, and at least 32 GB free hard disk space. Express your answer in terms of u : “You can upgrade your operating system,” b_{32} : “You have a 32-bit processor,” b_{64} : “You have a 64-bit processor,” g_1 : “Your processor runs at 1 GHz or faster,” g_2 : “Your processor runs at 2 GHz or faster,” r_1 : “Your processor has at least 1 GB RAM,” r_2 : “Your processor has at least 2 GB RAM,” h_{16} : “You have at least 16 GB free hard disk space,” and h_{32} : “You have at least 32 GB free hard disk space.”
7. Express these system specifications using the propositions p : “The message is scanned for viruses” and q : “The message was sent from an unknown system” together with logical connectives (including negations).
 - a) “The message is scanned for viruses whenever the message was sent from an unknown system.”
 - b) “The message was sent from an unknown system but it was not scanned for viruses.”
 - c) “It is necessary to scan the message for viruses whenever it was sent from an unknown system.”
 - d) “When a message is not sent from an unknown system it is not scanned for viruses.”
8. Express these system specifications using the propositions p : “The user enters a valid password,” q : “Access is granted,” and r : “The user has paid the subscription fee”

and logical connectives (including negations).

- a) “The user has paid the subscription fee, but does not enter a valid password.”
- b) “Access is granted whenever the user has paid the subscription fee and enters a valid password.”
- c) “Access is denied if the user has not paid the subscription fee.”

d) “If the user has not entered a valid password but has paid the subscription fee, then access is granted.” 9. Are these system specifications consistent? “The system is in multiuser state if and only if it is operating normally. If the system is operating normally, the kernel is functioning. The kernel is not functioning or the system is in interrupt mode. If the system is not in multiuser state, then it is in interrupt mode. The system is not in interrupt mode.”

- 10. Are these system specifications consistent? “Whenever the system software is being upgraded, users cannot access the file system. If users can access the file system, then they can save new files. If users cannot save new files, then the system software is not being upgraded.”
- 11. Are these system specifications consistent? “The router can send packets to the edge system only if it supports the new address space. For the router to support the new address space it is necessary that the latest software release be installed. The router can send packets to the edge system if the latest software release is installed, The router does not support the new address space.”
- 12. Are these system specifications consistent? “If the file system is not locked, then new messages will be queued. If the file system is not locked, then the system is functioning normally, and conversely. If new messages are not queued, then they will be sent to the message buffer. If the file system is not locked, then new messages will be sent to the message buffer. New messages will not be sent to the message buffer.”
- 13. What Boolean search would you use to look for Web pages about beaches in New Jersey? What if you wanted to find Web pages about beaches on the isle of Jersey (in the English Channel)?
- 14. What Boolean search would you use to look for Web pages about hiking in West Virginia? What if you wanted to find Web pages about hiking in Virginia, but not in West Virginia?

* 15. Each inhabitant of a remote village always tells the truth or always lies. A villager will give only a “Yes” or a “No” response to a question a tourist asks. Suppose you are a tourist visiting this area and come to a fork in the road. One branch leads to the ruins you want to visit; the other branch leads deep into the jungle. A villager is standing at the fork in the road. What one question can you ask the villager to determine which branch to take?

- 16. An explorer is captured by a group of cannibals. There are two types of cannibals—those who always tell the truth and those who always lie. The cannibals will barbecue the explorer unless he can determine whether a particular cannibal always lies or always tells the truth. He is allowed to ask the cannibal exactly one

question.

- a) Explain why the question “Are you a liar?” does not work.
- b) Find a question that the explorer can use to determine whether the cannibal always lies or always tells the truth.

- 17. When three professors are seated in a restaurant, the hostess asks them: “Does everyone want coffee?” The first professor says: “I do not know.” The second professor then says: “I do not know.” Finally, the third professor says: “No, not everyone wants coffee.” The hostess comes back and gives coffee to the professors who want it. How did she figure out who wanted coffee?
- 18. When planning a party you want to know whom to invite. Among the people you would like to invite are three touchy friends. You know that if Jasmine attends, she will

1.2 Applications of Propositional Logic 23

become unhappy if Samir is there, Samir will attend only if Kanti will be there, and Kanti will not attend unless Jasmine also does. Which combinations of these three friends can you invite so as not to make someone unhappy?

Exercises 19–23 relate to inhabitants of the island of knights and knaves created by Smullyan, where knights always tell the truth and knaves always lie. You encounter two people, **A** and **B**. Determine, if possible, what **A** and **B** are if they address you in the ways described. If you cannot determine what these two people are, can you draw any conclusions?

- 19. **A** says “At least one of us is a knave” and **B** says nothing.

20. **A** says “The two of us are both knights” and **B** says “**A** is a knave.”

21. **A** says “I am a knave or **B** is a knight” and **B** says nothing. 22. Both **A** and **B** say “I am a knight.”

23. **A** says “We are both knaves” and **B** says nothing.

Exercises 24–31 relate to inhabitants of an island on which there are three kinds of people: knights who always tell the truth, knaves who always lie, and spies (called normals by Smullyan [Sm78]) who can either lie or tell the truth. You encounter three people, **A**, **B**, and **C**. You know one of these people is a knight, one is a knave, and one is a spy. Each of the three people knows the type of person each of the other two is. For each of these situations, if possible, determine whether there is a unique solution and determine who the knave, knight, and spy are. When there is no unique solution, list all possible solutions or state that there are no solutions.

24. **A** says “**C** is the knave,” **B** says, “**A** is the knight,” and **C** says “I am the spy.”

25. **A** says “I am the knight,” **B** says “I am the knave,” and **C** says “**B** is the knight.”

26. **A** says “I am the knave,” **B** says “I am the knave,” and **C** says “I am the knave.”

27. **A** says “I am the knight,” **B** says “**A** is telling the truth,” and **C** says “I am the spy.”

28. **A** says “I am the knight,” **B** says, “**A** is not the knave,” and **C** says “**B** is not the knave.”

29. *A* says “I am the knight,” *B* says “I am the knight,” and *C* says “I am the knight.”

30. *A* says “I am not the spy,” *B* says “I am not the spy,” and *C* says “*A* is the spy.”

31. *A* says “I am not the spy,” *B* says “I am not the spy,” and *C* says “I am not the spy.”

Exercises 32–38 are puzzles that can be solved by translating statements into logical expressions and reasoning from these expressions using truth tables.

32. The police have three suspects for the murder of Mr. Cooper: Mr. Smith, Mr. Jones, and Mr. Williams. Smith, Jones, and Williams each declare that they did not kill Cooper. Smith also states that Cooper was a friend of Jones and that Williams disliked him. Jones also states that he did not know Cooper and that he was out of town the day Cooper was killed. Williams also states that he

241 / The Foundations: Logic and Proofs

saw both Smith and Jones with Cooper the day of the killing and that either Smith or Jones must have killed him. Can you determine who the murderer was if a) one of the three men is guilty, the two innocent men are telling the truth, but the statements of the guilty man may or may not be true?

b) innocent men do not lie?

33. Steve would like to determine the relative salaries of three coworkers using two facts. First, he knows that if Fred is not the highest paid of the three, then Janice is. Second, he knows that if Janice is not the lowest paid, then Maggie is paid the most. Is it possible to determine the relative salaries of Fred, Maggie, and Janice from what Steve knows? If so, who is paid the most and who the least? Explain your reasoning.

34. Five friends have access to a chat room. Is it possible to determine who is chatting if the following information is known? Either Kevin or Heather, or both, are chatting. Either Randy or Vijay, but not both, are chatting. If Abby is chatting, so is Randy. Vijay and Kevin are either both chatting or neither is. If Heather is chatting, then so are Abby and Kevin. Explain your reasoning.

35. A detective has interviewed four witnesses to a crime. From the stories of the witnesses the detective has concluded that if the butler is telling the truth then so is the cook; the cook and the gardener cannot both be telling the truth; the gardener and the handyman are not both lying; and if the handyman is telling the truth then the

I did it.”

a) If the authorities also know that exactly one of the

four suspects is telling the truth, who did it? Explain

your reasoning.

b) If the authorities also know that exactly one is lying, who did it? Explain your reasoning.

37. Suppose there are signs on the doors to two rooms. The

cook is lying. For each of the four witnesses, can the detective determine whether that person is telling the truth or lying? Explain your reasoning.

36. Four friends have been identified as suspects for an unauthorized access into a computer system. They have made statements to the investigating authorities. Alice said “Carlos did it.” John said “I did not do it.” Carlos said “Diana did it.” Diana said “Carlos lied when he said that

whose favorite drink is mineral water (which is one of the favorite drinks) given these clues: The Englishman lives in the red house. The Spaniard owns a dog. The Japanese man is a painter. The Italian drinks tea. The Norwegian lives in the first house on the left. The green house is immediately to the right of the white one. The photographer breeds snails. The diplomat lives in the yellow house. Milk is drunk in the middle house. The owner of the green house drinks coffee. The Norwegian’s house is next to the blue one. The violinist drinks orange juice. The fox is in a house next to that of the physician. The horse is in a house next to that of the diplomat. [Hint: Make a table where the rows represent the men and columns represent the color of their houses, their jobs, their pets, and their favorite drinks and use logical reasoning to determine the correct entries in the table.]

39. Freedonia has fifty senators. Each senator is either honest or corrupt. Suppose you know that at least one of the Freedonian senators is honest and that, given any two Freedonian senators, at least one is corrupt. Based on these facts, can you determine how many Freedonian senators are honest and how many are corrupt? If so, what is the answer?

40. Find the output of each of these combinatorial circuits.

a) *p*

q

b)

p

p

q

41. Find the output of each of these combinatorial circuits.

a)

p

q

r

b)

p

q

sign on the first door reads “In this room there is a lady, and in the other one there is a tiger”; and the sign on the

second door reads “In one of these rooms, there is a lady, and in one of them there is a tiger.” Suppose that you

know that one of these signs is true and the other is false. Behind which door is the lady? *

38. Solve this famous logic puzzle, attributed to Albert Einstein, and known as the **zebra puzzle**. Five men with different nationalities and with different jobs live in consecutive houses on a street. These houses are painted different colors. The men have different pets and have different favorite drinks. Determine who owns a zebra and

p

r

- 42.** Construct a combinatorial circuit using inverters, OR gates, and AND gates that produces the output $(p \wedge \neg r) \vee (\neg q \wedge r)$ from input bits p , q , and r .
- 43.** Construct a combinatorial circuit using inverters, OR gates, and AND gates that produces the output $((\neg p \vee \neg r) \wedge \neg q) \vee (\neg p \wedge (q \vee r))$ from input bits p , q , and r .

Propositional Equivalences

Introduction

An important type of step used in a mathematical argument is the replacement of a statement with another statement with the same truth value. Because of this, methods that produce propositions with the same truth value as a given compound proposition are used extensively in the construction of mathematical arguments. Note that we will use the term “compound proposition” to refer to an expression formed from propositional variables using logical operators, such as $p \wedge q$.

We begin our discussion with a classification of compound propositions according to their possible truth values.

DEFINITION 1 A compound proposition that is always true, no matter what the truth values of the propositional variables that occur in it, is called a *tautology*. A compound proposition that is always

false is called a *contradiction*. A compound proposition that is neither a tautology nor a contradiction is called a *contingency*.

Tautologies and contradictions are often important in mathematical reasoning. Example 1 illustrates these types of compound propositions.

EXAMPLE 1 We can construct examples of tautologies and contradictions using just one propositional variable. Consider the truth tables of $p \vee \neg p$ and $p \wedge \neg p$, shown in Table 1. Because $p \vee \neg p$ is always true, it is a tautology. Because $p \wedge \neg p$ is always false, it is a contradiction.

Logical Equivalences

Compound propositions that have the same truth values in all possible cases are called **logically equivalent**. We can also define this notion as follows.

DEFINITION 2 The compound propositions p and q are called *logically equivalent* if $p \leftrightarrow q$ is a tautology. The notation $p \equiv q$ denotes that p and q are logically equivalent.

Remark: The symbol \equiv is not a logical connective, and $p \equiv q$ is not a compound proposition but rather is the statement that $p \leftrightarrow q$ is a tautology. The symbol \Leftrightarrow is sometimes used instead of \equiv to denote logical equivalence.

One way to determine whether two compound propositions are equivalent is to use a truth table. In particular, the compound propositions p and q are equivalent if and only if the columns

TABLE 1 Examples of a Tautology and a Contradiction.			
p	$\neg p$	$p \vee \neg p$	$p \wedge \neg p$
T	F	T	F
F	T	T	F

26 1 / The Foundations: Logic and Proofs

TABLE 2 De Morgan's Laws.
$\neg(p \wedge q) \equiv \neg p \vee \neg q$ $\neg(p \vee q) \equiv \neg p \wedge \neg q$

giving their truth values agree. Example 2 illustrates this method to establish an extremely important and useful logical equivalence, namely, that of $\neg(p \vee q)$ with $\neg p \wedge \neg q$. This logical equivalence is one of the two **De Morgan laws**, shown in Table 2, named after the English mathematician Augustus De Morgan, of the mid-nineteenth century.

EXAMPLE 2 Show that $\neg(p \vee q)$ and $\neg p \wedge \neg q$ are logically equivalent.

Solution: The truth tables for these compound propositions are displayed in Table 3. Because the truth values of the compound propositions $\neg(p \vee q)$ and $\neg p \wedge \neg q$ agree for all possible combinations of the truth values of p and q , it follows that $\neg(p \vee q) \leftrightarrow (\neg p \wedge \neg q)$ is a tautology and that these compound propositions are logically equivalent. .

TABLE 3 Truth Tables for $\neg(p \vee q)$ and $\neg p \wedge \neg q$.					
$p \ q$	$p \vee q$	$\neg(p \vee q)$	$\neg p$	$\neg q$	$\neg p \wedge \neg q$
T T	T	F	F	F	F
T F	T	F	F	T	F
F T	T	F	T	F	F
F F	F	T	T	T	T

EXAMPLE 3 Show that $p \rightarrow q$ and $\neg p \vee q$ are logically equivalent.

Solution: We construct the truth table for these compound propositions in Table 4. Because the truth values of $\neg p \vee q$ and $p \rightarrow q$ agree, they are logically equivalent. .

TABLE 4 Truth Tables for $\neg p \vee q$ and $p \rightarrow q$.			
$p \ q$	$\neg p$	$\neg p \vee q$	$p \rightarrow q$
T T	F	T	T
T F	F	F	F
F T	T	T	T
F F	T	T	T

We will now establish a logical equivalence of two compound propositions involving three different propositional variables p , q , and r . To use a truth table to establish such a logical equivalence, we need eight rows, one for each possible combination of truth values of these three variables. We symbolically represent these combinations by listing the truth values of p , q , and r , respectively. These eight combinations of truth values are TTT, TTF, TFT, TFF, FTT, FTF, FFT, and FFF; we use this order when we display the rows of the truth table. Note that we need to double the number of rows in the truth tables we use to show that compound propositions are equivalent for each additional propositional variable, so that 16 rows are needed to establish the logical equivalence of two compound propositions involving four propositional variables, and so on. In general, 2^n rows are required if a compound proposition involves n propositional variables.

TABLE 5 A Demonstration That $p \vee (q \wedge r)$ and $(p \vee q) \wedge (p \vee r)$ Are Logically Equivalent.					
pqr	$q \wedge r$	$p \vee (q \wedge r)$	$p \vee q$	$p \vee r$	$(p \vee q) \wedge (p \vee r)$
TTT	T	T	T	T	T
TTF	F	T	T	T	T
TFT	F	T	T	T	T
TFF	F	T	T	T	T
FTT	T	T	T	T	T
FTF	F	F	T	F	F
FFT	F	F	F	T	F
FFF	F	F	F	F	F

EXAMPLE 4 Show that $p \vee (q \wedge r)$ and $(p \vee q) \wedge (p \vee r)$ are logically equivalent. This is the *distributive law* of disjunction over conjunction.

Solution: We construct the truth table for these compound propositions in Table 5. Because the truth values of $p \vee (q \wedge r)$ and $(p \vee q) \wedge (p \vee r)$ agree, these compound propositions are logically equivalent. .

The identities in Table 6 are a special case of Boolean algebra identities found in Table 5 of Section 12.1. See Table 1 in Section 2.2 for analogous set identities.

Table 6 contains some important equivalences. In these equivalences, **T** denotes the compound proposition that is always true and **F** denotes the compound proposition that is always

TABLE 6 Logical Equivalences.	
Equivalence	
$p \wedge \mathbf{T} \equiv p$ $p \vee \mathbf{F} \equiv p$	
$p \vee \mathbf{T} \equiv \mathbf{T}$ $p \wedge \mathbf{F} \equiv \mathbf{F}$	
$p \vee p \equiv p$ $p \wedge p \equiv p$	

$\neg(\neg p) \equiv p$	Double negation
$p \vee q \equiv q \vee p$ $p \wedge q \equiv q \wedge p$	Commutative
$(p \vee q) \vee r \equiv p \vee (q \vee r)$ $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$	Associative
$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ $(p \wedge r) \vee (q \wedge r) \equiv (p \vee q) \wedge r$	Distributive
$\neg(p \wedge q) \equiv \neg p \vee \neg q$ $\neg(p \vee q) \equiv \neg p \wedge \neg q$	De Morgan's
$p \vee (p \wedge q) \equiv p$ $p \wedge (p \vee q) \equiv p$	Absorption
$p \vee \neg p \equiv \mathbf{T}$ $p \wedge \neg p \equiv \mathbf{F}$	Negation

TABLE 7 Logical Equivalences Involving Conditional Statements.

$$\begin{aligned}
p \rightarrow q &\equiv \neg p \vee q \\
p \rightarrow q &\equiv \neg q \rightarrow \neg p \\
p \vee q &\equiv \neg p \rightarrow q \\
p \wedge q &\equiv \neg(p \rightarrow \neg q) \\
\neg(p \rightarrow q) &\equiv p \wedge \neg q \\
(p \rightarrow q) \wedge (p \rightarrow r) &\equiv p \rightarrow (q \wedge r) \\
(p \rightarrow r) \wedge (q \rightarrow r) &\equiv (p \vee q) \rightarrow r \\
(p \rightarrow q) \vee (p \rightarrow r) &\equiv p \rightarrow (q \vee r)
\end{aligned}$$

$$\wedge q) \rightarrow r$$

TABLE 8 Logical Equivalences Involving Biconditional Statements.

$$\begin{aligned}
p \leftrightarrow q &\equiv (p \rightarrow q) \wedge (q \rightarrow p) \\
p \leftrightarrow q &\equiv \neg p \leftrightarrow \neg q \\
p \leftrightarrow q &\equiv (p \wedge q) \vee (\neg p \wedge \neg q) \\
\neg(p \leftrightarrow q) &\equiv p \leftrightarrow \neg q
\end{aligned}$$

false. We also display some useful equivalences for compound propositions involving conditional statements and biconditional statements in Tables 7 and 8, respectively. The reader is asked to verify the equivalences in Tables 6–8 in the exercises.

The associative law for disjunction shows that the expression $p \vee q \vee r$ is well defined, in the sense that it does not matter whether we first take the disjunction of p with q and then the disjunction of $p \vee q$ with r , or if we first take the disjunction of q and r and then take the disjunction of p with $q \vee r$. Similarly, the expression $p \wedge q \wedge r$ is well defined. By extending this reasoning, it follows that $p_1 \vee p_2 \vee \cdots \vee p_n$ and $p_1 \wedge p_2 \wedge \cdots \wedge p_n$ are well defined whenever p_1, p_2, \dots, p_n are propositions.

Furthermore, note that De Morgan's laws extend to

$$\neg(p_1 \vee p_2 \vee \cdots \vee p_n) \equiv (\neg p_1 \wedge \neg p_2 \wedge \cdots \wedge \neg p_n)$$

and

$$\neg(p_1 \wedge p_2 \wedge \cdots \wedge p_n) \equiv (\neg p_1 \vee \neg p_2 \vee \cdots \vee \neg p_n).$$

We will sometimes use the notation $\bigvee_{j=1}^n p_j$ for $p_1 \vee p_2 \vee \cdots \vee p_n$ and $\bigwedge_{j=1}^n p_j$ for $p_1 \wedge p_2 \wedge \cdots \wedge p_n$. Using this notation, the extended version of De Morgan's laws can be

written concisely as $\neg \bigvee_{j=1}^n p_j \equiv \bigwedge_{j=1}^n \neg p_j$ and $\neg \bigwedge_{j=1}^n p_j \equiv \bigvee_{j=1}^n \neg p_j$. (Methods for proving these identities will be given in Section 5.1.)

Using De Morgan's Laws

The two logical equivalences known as De Morgan's laws are particularly important. They tell us how to negate conjunctions and how to negate disjunctions. In particular, the equivalence $\neg(p \vee q) \equiv \neg p \wedge \neg q$ tells us that the negation of a disjunction is formed by taking the conjunction of the negations of the component propositions. Similarly, the equivalence $\neg(p \wedge q) \equiv \neg p \vee \neg q$ tells us that the negation of a conjunction is formed by taking the disjunction of the negations of the component propositions. Example 5 illustrates the use of De Morgan's laws.

EXAMPLE 5 Use De Morgan's laws to express the negations of "Miguel has a cellphone and he has a laptop computer" and "Heather will go to the concert or Steve will go to the concert."

Solution: Let p be "Miguel has a cellphone" and q be "Miguel has a laptop computer." Then "Miguel has a cellphone and he has a laptop computer" can be represented by $p \wedge q$. By the first of De Morgan's laws, $\neg(p \wedge q)$ is equivalent to $\neg p \vee \neg q$. Consequently, we can express the negation of our original statement as "Miguel does not have a cellphone or he does not have a laptop computer."

Let r be "Heather will go to the concert" and s be "Steve will go to the concert." Then "Heather will go to the concert or Steve will go to the concert" can be represented by $r \vee s$. By the second of De Morgan's laws, $\neg(r \vee s)$ is equivalent to $\neg r \wedge \neg s$. Consequently, we can express the negation of our original statement as "Heather will not go to the concert and Steve will not go to the concert."

Constructing New Logical Equivalences

The logical equivalences in Table 6, as well as any others that have been established (such as those shown in Tables 7 and 8), can be used to construct additional logical equivalences. The reason for this is that a proposition in a compound proposition can be replaced by a compound proposition that is logically equivalent to it without changing the truth value of the original compound proposition. This technique is illustrated in Examples 6–8, where we also use the fact that if p and q are logically equivalent and q and r are logically equivalent, then p and r are logically equivalent (see Exercise 56).

EXAMPLE 6 Show that $\neg(p \rightarrow q)$ and $p \wedge \neg q$ are logically equivalent.

Solution: We could use a truth table to show that these compound propositions are equivalent (similar to what we did in Example 4). Indeed, it would not be hard to do so. However, we want to illustrate how to use logical identities that we already know to establish new logical identities, something that is of practical importance for establishing equivalences of compound propositions with a large number of variables. So, we will establish this equivalence by developing a series of

AUGUSTUS DE MORGAN (1806–1871) Augustus De Morgan was born in India, where his father was a colonel in the Indian army. De Morgan's family moved to England when he was 7 months old. He attended private schools, where in his early teens he developed a strong interest in mathematics. De Morgan studied at Trinity College, Cambridge, graduating in 1827. Although he considered medicine or law, he decided on mathematics for his career. He won a position at University College, London, in 1828, but resigned after the college dismissed a fellow professor without giving reasons. However, he resumed this position in 1836 when his successor died, remaining until 1866.

De Morgan was a noted teacher who stressed principles over techniques. His students included many famous mathematicians, including Augusta Ada, Countess of Lovelace, who was Charles Babbage's collaborator in his work on computing machines (see page 31 for biographical notes on Augusta Ada). (De Morgan cautioned the countess against studying too much mathematics, because it might interfere with her childbearing abilities!)

De Morgan was an extremely prolific writer, publishing more than 1000 articles in more than 15 periodicals. De Morgan also wrote textbooks on many subjects, including logic, probability, calculus, and algebra. In 1838 he presented what was perhaps the first clear explanation of an important proof technique known as *mathematical induction* (discussed in Section 5.1 of this text), a term he coined. In the 1840s De Morgan made fundamental contributions to the development of symbolic logic. He invented notations that helped him prove propositional equivalences, such as the laws that are named after him. In 1842 De Morgan presented what is considered to be the first precise definition of a limit and developed new tests for convergence of infinite series. De Morgan was also interested in the history of mathematics and wrote biographies of Newton and Halley.

In 1837 De Morgan married Sophia Frend, who wrote his biography in 1882. De Morgan's research, writing, and teaching left little time for his family or social life. Nevertheless, he was noted for his kindness, humor, and wide range of knowledge.

logical equivalences, using one of the equivalences in Table 6 at a time, starting with $\neg(p \rightarrow q)$ and ending with $p \wedge \neg q$. We have the following equivalences.

$$\begin{aligned}\neg(p \rightarrow q) &\equiv \neg(\neg p \vee q) \text{ by Example 3} \\ &\equiv \neg(\neg p) \wedge \neg q \text{ by the second De Morgan law} \\ &\equiv p \wedge \neg q \text{ by the double negation law}\end{aligned}$$

EXAMPLE 7 Show that $\neg(p \vee (\neg p \wedge q))$ and $\neg p \wedge \neg q$ are logically equivalent by developing a series of logical equivalences.

Solution: We will use one of the equivalences in Table 6 at a time, starting with $\neg(p \vee (\neg p \wedge q))$ and ending with $\neg p \wedge \neg q$. (Note: we could also easily establish this equivalence using a truth table.) We have the following equivalences.

$$\begin{aligned}\neg(p \vee (\neg p \wedge q)) &\equiv \neg p \wedge \neg(\neg p \wedge q) \text{ by the second De Morgan law} \\ &\equiv \neg p \wedge [\neg(\neg p) \vee \neg q] \text{ by the first De Morgan law} \\ &\equiv \neg p \wedge (p \vee \neg q) \text{ by the double negation law} \\ &\equiv (\neg p \wedge p) \vee (\neg p \wedge \neg q) \text{ by the second distributive law} \\ &\equiv \mathbf{F} \vee (\neg p \wedge \neg q) \text{ because } \neg p \wedge p \equiv \mathbf{F} \\ &\equiv (\neg p \wedge \neg q) \vee \mathbf{F} \text{ by the commutative law for disjunction} \\ &\equiv \neg p \wedge \neg q \text{ by the identity law for } \mathbf{F}\end{aligned}$$

Consequently $\neg(p \vee (\neg p \wedge q))$ and $\neg p \wedge \neg q$ are logically equivalent.

EXAMPLE 8 Show that $(p \wedge q) \rightarrow (p \vee q)$ is a tautology.

Solution: To show that this statement is a tautology, we will use logical equivalences to demonstrate that it is logically equivalent to **T**. (Note: This could also be done using a truth table.)

$$\begin{aligned}(p \wedge q) \rightarrow (p \vee q) &\equiv \neg(p \wedge q) \vee (p \vee q) \text{ by Example 3} \\ &\equiv (\neg p \vee \neg q) \vee (p \vee q) \text{ by the first De Morgan law} \\ &\equiv (\neg p \vee p) \vee (\neg q \vee q) \text{ by the associative and commutative} \\ &\hspace{10em} \text{laws for disjunction} \\ &\equiv \mathbf{T} \vee \mathbf{T} \text{ by Example 1 and the commutative} \\ &\hspace{10em} \text{law for disjunction} \\ &\equiv \mathbf{T} \text{ by the domination law}\end{aligned}$$

Propositional Satisfiability

A compound proposition is **unsatisfiable** if there is an assignment of truth values to its variables that makes it true. When no such assignment exists, that is, when the compound proposition is false for all assignments of truth values to its variables, the compound proposition is **unsatisfiable**.

Note that a compound proposition is unsatisfiable if and only if its negation is true for all assignments of truth values to the variables, that is, if and only if its negation is a tautology. When we find a particular assignment of truth values that makes a compound proposition true, we have shown that it is satisfiable; such an assignment is called a **solution** of this particular

satisfiability problem. However, to show that a compound proposition is unsatisfiable, we need to show that *every* assignment of truth values to its variables makes it false. Although we can always use a truth table to determine whether a compound proposition is satisfiable, it is often more efficient not to, as Example 9 demonstrates.

EXAMPLE 9 Determine whether each of the compound propositions $(p \vee \neg q) \wedge (q \vee \neg r) \wedge (r \vee \neg p)$, $(p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$, and $(p \vee \neg q) \wedge (q \vee \neg r) \wedge (r \vee \neg p) \wedge (p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$ is satisfiable.

Solution: Instead of using truth table to solve this problem, we will reason about truth values. Note that $(p \vee \neg q) \wedge (q \vee \neg r) \wedge (r \vee \neg p)$ is true when the three variable p , q , and r have the same truth value (see Exercise 40 of Section 1.1). Hence, it is satisfiable as there is at least one assignment of truth values for p , q , and r that makes it true. Similarly, note that $(p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$ is true when at least one of p , q , and r is true and at least one is false (see Exercise 41 of Section 1.1). Hence, $(p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$ is satisfiable, as there is at least one assignment of truth values for p , q , and r that makes it true.

Finally, note that for $(p \vee \neg q) \wedge (q \vee \neg r) \wedge (r \vee \neg p) \wedge (p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$ to be true, $(p \vee \neg q) \wedge (q \vee \neg r) \wedge (r \vee \neg p)$ and $(p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$ must both be true. For the first to be true, the three variables must have the same truth values, and for the second to be true, at least one of three variables must be true and at least one must be false. However, these conditions are contradictory. From these observations we conclude that no assignment of truth values to p , q , and r makes $(p \vee \neg q) \wedge (q \vee \neg r) \wedge (r \vee \neg p) \wedge (p \vee q \vee r) \wedge (\neg p \vee \neg q \vee \neg r)$ true. Hence, it is unsatisfiable. .

AUGUSTA ADA, COUNTESS OF LOVELACE (1815–1852) Augusta Ada was the only child from the marriage of the famous poet Lord Byron and Lady Byron, Annabella Millbanke, who separated when Ada was 1 month old, because of Lord Byron’s scandalous affair with his half sister. The Lord Byron had quite a reputation, being described by one of his lovers as “mad, bad, and dangerous to know.” Lady Byron was noted for her intellect and had a passion for mathematics; she was called by Lord Byron “The Princess of Parallelograms.” Augusta was raised by her mother, who encouraged her intellectual talents especially in music and mathematics, to counter what Lady Byron considered dangerous poetic tendencies. At this time, women were not allowed to attend universities and could not join learned societies. Nevertheless, Augusta pursued her mathematical studies independently and with mathematicians, including William Frend. She was also encouraged by another female

mathematician, Mary Somerville, and in 1834 at a dinner party hosted by Mary Somerville, she learned about Charles Babbage’s ideas for a calculating machine, called the Analytic Engine. In 1838 Augusta Ada married Lord King, later elevated to Earl of Lovelace. Together they had three children.

Augusta Ada continued her mathematical studies after her marriage. Charles Babbage had continued work on his Analytic Engine and lectured on this in Europe. In 1842 Babbage asked Augusta Ada to translate an article in French describing Babbage’s invention. When Babbage saw her translation, he suggested she add her own notes, and the resulting work was three times the length of the original. The most complete accounts of the Analytic Engine are found in Augusta Ada’s notes. In her notes, she compared the working of the Analytic Engine to that of the Jacquard loom, with Babbage’s punch cards analogous to the cards used to create patterns on the loom. Furthermore, she recognized the promise of the machine as a general purpose computer much better than Babbage did. She stated that the “engine is the material expression of any indefinite function of any degree of generality and complexity.” Her notes on the Analytic Engine anticipate many future developments, including computer-generated music. Augusta Ada published her writings under her initials A.A.L. concealing her identity as a woman as did many women at a time when women were not considered to be the intellectual equals of men. After 1845 she and Babbage worked toward the development of a system to predict horse races. Unfortunately, their system did not work well, leaving Augusta Ada heavily in debt at the time of her death at an unfortunately young age from uterine cancer.

In 1953 Augusta Ada’s notes on the Analytic Engine were republished more than 100 years after they were written, and after they had been long forgotten. In his work in the 1950s on the capacity of computers to think (and his famous Turing Test), Alan Turing responded to Augusta Ada’s statement that “The Analytic Engine has no pretensions whatever to originate anything. It can do whatever we know how to order it to perform.” This “dialogue” between Turing and Augusta Ada is still the subject of controversy. Because of her fundamental contributions to computing, the programming language Ada is named in honor of the Countess of Lovelace.

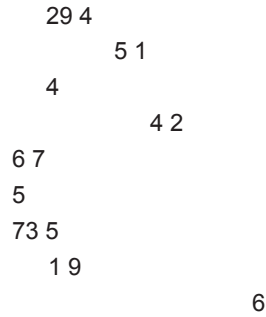


FIGURE 1 A 9×9 Sudoku puzzle.

Applications of Satisfiability

Many problems, in diverse areas such as robotics, software testing, computer-aided design, machine vision, integrated circuit design, computer networking, and genetics, can be modeled in terms of propositional satisfiability. Although most of these applications are beyond the scope of this book, we will study one application here. In particular, we will show how to use propositional satisfiability to model Sudoku puzzles.

SUDOKU A **Sudoku puzzle** is represented by a 9×9 grid made up of nine 3×3 subgrids, known as **blocks**, as shown in Figure 1. For each puzzle, some of the 81 cells, called **givens**, are assigned one of the numbers 1, 2, ..., 9, and the other cells are blank. The puzzle is solved by assigning a number to each blank cell so that every row, every column, and every one of the nine 3×3 blocks contains each of the nine possible numbers. Note that instead of using a 9×9 grid, Sudoku puzzles can be based on $n^2 \times n^2$ grids, for any positive integer n , with the $n^2 \times n^2$ grid made up of $n^2 \times n^2$ subgrids.

The popularity of Sudoku dates back to the 1980s when it was introduced in Japan. It took 20 years for Sudoku to spread to rest of the world, but by 2005, Sudoku puzzles were a worldwide craze. The name Sudoku is short for the Japanese *suuji wa dokushin ni kagiru*, which means “the digits must remain single.” The modern game of Sudoku was apparently designed in the late 1970s by an American puzzle designer. The basic ideas of Sudoku date back even further; puzzles printed in French newspapers in the 1890s were quite similar, but not identical, to modern Sudoku.

Sudoku puzzles designed for entertainment have two additional important properties. First, they have exactly one solution. Second, they can be solved using reasoning alone, that is, without resorting to searching all possible assignments of numbers to the cells. As a Sudoku puzzle is solved, entries in blank cells are successively determined by already known values. For instance, in the grid in Figure 1, the number 4 must appear in exactly one cell in the second row. How can we determine which of the seven blank cells it must appear? First, we observe that 4 cannot appear in one of the first three cells or in one of the last three cells of this row, because it already appears in another cell in the block each of these cells is in. We can also see that 4 cannot appear in the fifth cell in this row, as it already appears in the fifth column in the fourth row. This means that 4 must appear in the sixth cell of the second row.

Many strategies based on logic and mathematics have been devised for solving Sudoku puzzles (see [Da10], for example). Here, we discuss one of the ways that have been developed for solving Sudoku puzzles with the aid of a computer, which depends on modeling the puzzle as a propositional satisfiability problem. Using the model we describe, particular Sudoku puzzles can be solved using software developed to solve satisfiability problems. Currently, Sudoku puzzles can be solved in less than 10 milliseconds this way. It should be noted that there are many other approaches for solving Sudoku puzzles via computers using other techniques.

To encode a Sudoku puzzle, let $p(i, j, n)$ denote the proposition that is true when the number n is in the cell in the i th row and j th column. There are $9 \times 9 \times 9 = 729$ such propositions, as i, j , and n all range from 1 to 9. For example, for the puzzle in Figure 1, the number 6 is given as the value in the fifth row and first column. Hence, we see that $p(5, 1, 6)$ is true, but $p(5, j, 6)$ is false for $j = 2, 3, \dots, 9$.

Given a particular Sudoku puzzle, we begin by encoding each of the given values. Then, we construct compound propositions that assert that every row contains every number, every column contains every number, every 3×3 block contains every number, and each cell contains no more than one number. It follows, as the reader should verify, that the Sudoku puzzle is solved by finding an assignment of truth values to the 729 propositions $p(i, j, n)$ with i, j , and n each ranging from 1 to 9 that makes the conjunction of all these compound propositions true. After listing these assertions, we will explain how to construct the assertion that every row contains every integer from 1 to 9. We will leave the construction of the other assertions that every column contains every number and each of the nine 3×3 blocks contains every number to the exercises.

For each cell with a given value, we assert $p(i, j, n)$ when the cell in row i and column j has the given value n .

We assert that every row contains every number:

$$\bigwedge_{i=1}^9 \bigwedge_{n=1}^9 \bigwedge_{j=1}^9 p(i, j, n)$$

We assert that every column contains every number:

$$\bigwedge_{j=1}^9 \bigwedge_{n=1}^9 \bigwedge_{i=1}^9 p(i, j, n)$$

We assert that each of the nine 3×3 blocks contains every number:

It is tricky setting up the two inner indices so that all nine cells in each square block are examined.

$$\bigwedge_{r=0}^2 \bigwedge_{s=0}^2 \bigwedge_{n=1}^9 \bigwedge_{i=1}^3 \bigwedge_{j=1}^3 p(3r + i, 3s + j, n)$$

To assert that no cell contains more than one number, we take the conjunction over all values of n , i , and j where each variable ranges from 1 to 9 and $n \neq n$ of $p(i, j, n) \rightarrow \neg p(i, j, n)$.

We now explain how to construct the assertion that every row contains every number.

First, to assert that row i contains the number n , we form $\bigwedge_{j=1}^9 p(i, j, n)$. To assert that row i contains all n numbers, we form the conjunction of these disjunctions over all nine possible values of n , giving us $\bigwedge_{n=1}^9 \bigwedge_{j=1}^9 p(i, j, n)$. Finally, to assert that every row contains every number, we take the conjunction of $\bigwedge_{n=1}^9 \bigwedge_{j=1}^9 p(i, j, n)$ over all nine rows. This gives us $\bigwedge_{i=1}^9 \bigwedge_{n=1}^9 \bigwedge_{j=1}^9 p(i, j, n)$.

$\prod_{n=1}^9 \prod_{j=1}^9 p(i, j, n)$. (Exercises 65 and 66 ask for explanations of the assertions that every column contains every number and that each of the nine 3×3 blocks contains every number.)

Given a particular Sudoku puzzle, to solve this puzzle we can find a solution to the satisfiability problems that asks for a set of truth values for the 729 variables $p(i, j, n)$ that makes the conjunction of all the listed assertions true.

Solving Satisfiability Problems

A truth table can be used to determine whether a compound proposition is satisfiable, or equivalently, whether its negation is a tautology (see Exercise 60). This can be done by hand for a compound proposition with a small number of variables, but when the number of variables grows, this becomes impractical. For instance, there are $2^{20} = 1,048,576$ rows in the truth table for a compound proposition with 20 variables. Clearly, you need a computer to help you determine, in this way, whether a compound proposition in 20 variables is satisfiable.

When many applications are modeled, questions concerning the satisfiability of compound propositions with hundreds, thousands, or millions of variables arise. Note, for example, that when there are 1000 variables, checking every one of the 2^{1000} (a number with more than 300 decimal digits) possible combinations of truth values of the variables in a compound proposition cannot be done by a computer in even trillions of years. No procedure is known that a computer can follow to determine in a reasonable amount of time whether an arbitrary compound proposition in such a large number of variables is satisfiable. However, progress has been made developing methods for solving the satisfiability problem for the particular types of compound propositions that arise in practical applications, such as for the solution of Sudoku puzzles. Many computer programs have been developed for solving satisfiability problems which have practical use. In our discussion of the subject of algorithms in Chapter 3, we will discuss this question further. In particular, we will explain the important role the propositional satisfiability problem plays in the study of the complexity of algorithms.

Exercises

1. Use truth tables to verify these equivalences.

$$\begin{aligned} \text{a) } p \wedge \mathbf{T} &\equiv p & \text{b) } p \vee \mathbf{F} &\equiv p & \text{c) } p \wedge \mathbf{F} &\equiv \mathbf{F} & \text{d) } \\ p \vee \mathbf{T} &\equiv \mathbf{T} & \text{e) } p \vee p &\equiv p & \text{f) } p \wedge p &\equiv p \end{aligned}$$

2. Show that $\neg(\neg p)$ and p are logically equivalent.

3. Use truth tables to verify the commutative laws

$$\text{a) } p \vee q \equiv q \vee p. \quad \text{b) } p \wedge q \equiv q \wedge p.$$

4. Use truth tables to verify the associative laws
a) $(p \vee q) \vee r \equiv p \vee (q \vee r)$.

$$\text{b) } (p \wedge q) \wedge r \equiv p \wedge (q \wedge r).$$

5. Use a truth table to verify the distributive law

$$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r).$$

6. Use a truth table to verify the first De Morgan law

$$\neg(p \wedge q) \equiv \neg p \vee \neg q.$$

7. Use De Morgan's laws to find the negation of each of the following statements.

a) Jan is rich and happy.

b) Carlos will bicycle or run tomorrow.



HENRY MAURICE SHEFFER (1883–1964) Henry Maurice Sheffer, born to Jewish parents in the western Ukraine, emigrated to the United States in 1892 with his parents and six siblings. He studied at the Boston Latin School before entering Harvard, where he completed his undergraduate degree in 1905, his master's in 1907, and his Ph.D. in philosophy in 1908. After holding a postdoctoral position at Harvard, Henry traveled to Europe on a fellowship. Upon returning to the United States, he became an academic nomad, spending one year each at the University of Washington, Cornell, the University of Minnesota, the University of Missouri, and City College in New York. In 1916 he returned to Harvard as a faculty member in the philosophy department. He remained at Harvard until his retirement in 1952.

Sheffer introduced what is now known as the Sheffer stroke in 1913; it became well known only after its use in the 1925 edition of Whitehead and Russell's *Principia Mathematica*. In this same edition Russell wrote that Sheffer had invented a powerful method that could be used to simplify the *Principia*. Because of this comment, Sheffer was something of a mystery man to logicians, especially because Sheffer, who published little in his career, never published the details of this method, only describing it in mimeographed notes and in a brief published abstract.

Sheffer was a dedicated teacher of mathematical logic. He liked his classes to be small and did not like auditors. When strangers appeared in his classroom, Sheffer would order them to leave, even his colleagues or distinguished guests visiting Harvard. Sheffer was barely five feet tall; he was noted for his wit and vigor, as well as for his nervousness and irritability.

Although widely liked, he was quite lonely. He is noted for a quip he spoke at his retirement: “Old professors never die, they just become emeriti.” Sheffer is also credited with coining the term “Boolean algebra” (the subject of Chapter 12 of this text). Sheffer was briefly married and lived most of his later life in small rooms at a hotel packed with his logic books and vast files of slips of paper he used to jot down his ideas. Unfortunately, Sheffer suffered from severe depression during the last two decades of his life.

28. Show that $p \leftrightarrow q$ and $\neg p \leftrightarrow \neg q$ are logically equivalent.

1.3 Propositional Equivalences 35

- c) Mei walks or takes the bus to class.
- d) Ibrahim is smart and hard working.
- 8. Use De Morgan’s laws to find the negation of each of the following statements.
 - a) Kwame will take a job in industry or go to graduate school.
 - b) Yoshiko knows Java and calculus.
 - c) James is young and strong.
 - d) Rita will move to Oregon or Washington.
- 9. Show that each of these conditional statements is a tautology by using truth tables.
 - a) $(p \wedge q) \rightarrow p$ b) $p \rightarrow (p \vee q)$ c) $\neg p \rightarrow (p \rightarrow q)$
 - d) $(p \wedge q) \rightarrow (p \rightarrow q)$ e) $\neg(p \rightarrow q) \rightarrow p$ f) $\neg(p \rightarrow q) \rightarrow \neg q$
- 10. Show that each of these conditional statements is a tautology by using truth tables.
 - a) $[\neg p \wedge (p \vee q)] \rightarrow q$
 - b) $[(p \rightarrow q) \wedge (q \rightarrow r)] \rightarrow (p \rightarrow r)$
 - c) $[p \wedge (p \rightarrow q)] \rightarrow q$
 - d) $[(p \vee q) \wedge (p \rightarrow r) \wedge (q \rightarrow r)] \rightarrow r$
- 11. Show that each conditional statement in Exercise 9 is a tautology without using truth tables.
- 12. Show that each conditional statement in Exercise 10 is a tautology without using truth tables.
- 13. Use truth tables to verify the absorption laws. a) $p \vee (p \wedge q) \equiv p$ b) $p \wedge (p \vee q) \equiv p$
- 14. Determine whether $(\neg p \wedge (p \rightarrow q)) \rightarrow \neg q$ is a tautology.
- 15. Determine whether $(\neg q \wedge (p \rightarrow q)) \rightarrow \neg p$ is a tautology.

Each of Exercises 16–28 asks you to show that two compound propositions are logically equivalent. To do this, either show that both sides are true, or that both sides are false, for exactly the same combinations of truth values of the propositional variables in these expressions (whichever is easier).

- 16. Show that $p \leftrightarrow q$ and $(p \wedge q) \vee (\neg p \wedge \neg q)$ are logically equivalent.
- 17. Show that $\neg(p \leftrightarrow q)$ and $p \leftrightarrow \neg q$ are logically equivalent.
- 18. Show that $p \rightarrow q$ and $\neg q \rightarrow \neg p$ are logically equivalent.
- 19. Show that $\neg p \leftrightarrow q$ and $p \leftrightarrow \neg q$ are logically equivalent.
- 20. Show that $\neg(p \oplus q)$ and $p \leftrightarrow q$ are logically equivalent.
- 21. Show that $\neg(p \leftrightarrow q)$ and $\neg p \leftrightarrow q$ are logically equivalent.
- 22. Show that $(p \rightarrow q) \wedge (p \rightarrow r)$ and $p \rightarrow (q \wedge r)$ are logically equivalent.
- 23. Show that $(p \rightarrow r) \wedge (q \rightarrow r)$ and $(p \vee q) \rightarrow r$ are logically equivalent.
- 24. Show that $(p \rightarrow q) \vee (p \rightarrow r)$ and $p \rightarrow (q \vee r)$ are logically equivalent.
- 25. Show that $(p \rightarrow r) \vee (q \rightarrow r)$ and $(p \wedge q) \rightarrow r$ are logically equivalent.
- 26. Show that $\neg p \rightarrow (q \rightarrow r)$ and $q \rightarrow (p \vee r)$ are logically equivalent.
- 27. Show that $p \leftrightarrow q$ and $(p \rightarrow q) \wedge (q \rightarrow p)$ are logically equivalent.

29. Show that $(p \rightarrow q) \wedge (q \rightarrow r) \rightarrow (p \rightarrow r)$ is a tautology.

30. Show that $(p \vee q) \wedge (\neg p \vee r) \rightarrow (q \vee r)$ is a tautology.

31. Show that $(p \rightarrow q) \rightarrow r$ and $p \rightarrow (q \rightarrow r)$ are not logically equivalent.

32. Show that $(p \wedge q) \rightarrow r$ and $(p \rightarrow r) \wedge (q \rightarrow r)$ are not logically equivalent.

33. Show that $(p \rightarrow q) \rightarrow (r \rightarrow s)$ and $(p \rightarrow r) \rightarrow (q \rightarrow s)$ are not logically equivalent.

The **dual** of a compound proposition that contains only the logical operators \vee , \wedge , and \neg is the compound proposition obtained by replacing each \vee by \wedge , each \wedge by \vee , each **T** by **F**, and each **F** by **T**. The dual of **s** is denoted by s^* .

34. Find the dual of each of these compound propositions.

- a) $p \vee \neg q$ b) $p \wedge (q \vee (r \wedge T))$ c) $(p \wedge \neg q) \vee (q \wedge F)$

35. Find the dual of each of these compound propositions.

- a) $p \wedge \neg q \wedge \neg r$ b) $(p \wedge q \wedge r) \vee s$ c) $(p \vee F) \wedge (q \vee T)$

36. When does $s^* = s$, where **s** is a compound proposition?

37. Show that $(s^*)^* = s$ when **s** is a compound proposition.

38. Show that the logical equivalences in Table 6, except for the double negation law, come in pairs, where each pair contains compound propositions that are duals of each other.

**

39. Why are the duals of two equivalent compound propositions also equivalent, where these compound propositions contain only the operators \wedge , \vee , and \neg ?

40. Find a compound proposition involving the propositional variables **p**, **q**, and **r** that is true when **p** and **q** are true and **r** is false, but is false otherwise. [Hint: Use a conjunction of each propositional variable or its negation.]

41. Find a compound proposition involving the propositional variables **p**, **q**, and **r** that is true when exactly two of **p**, **q**, and **r** are true and is false otherwise. [Hint: Form a disjunction of conjunctions. Include a conjunction for each combination of values for which the compound proposition is true. Each conjunction should include each of the three propositional variables or its negations.]

42. Suppose that a truth table in *n* propositional variables is specified. Show that a compound proposition with this truth table can be formed by taking the disjunction of conjunctions of the variables or their negations, with one conjunction included for each combination of values for which the compound proposition is true. The resulting compound proposition is said to be in **disjunctive normal form**.

A collection of logical operators is called **functionally complete** if every compound proposition is logically equivalent to a compound proposition involving only these logical operators.

Predicates and Quantifiers

Introduction

43. Show that \neg , \wedge , and \vee form a functionally complete collection of logical operators. [Hint: Use the fact that every compound proposition is logically equivalent to one in disjunctive normal form, as shown in Exercise 42.]
- 36 1 / The Foundations: Logic and Proofs

- *44. Show that \neg and \wedge form a functionally complete collection of logical operators. [Hint: First use a De Morgan law to show that $p \vee q$ is logically equivalent to $\neg(\neg p \wedge \neg q)$.]
- *45. Show that \neg and \vee form a functionally complete collection of logical operators.

The following exercises involve the logical operators *NAND* and *NOR*. The proposition $p \text{ NAND } q$ is true when either p or q , or both, are false; and it is false when both p and q are true. The proposition $p \text{ NOR } q$ is true when both p and q are false, and it is false otherwise. The propositions $p \text{ NAND } q$ and $p \text{ NOR } q$ are denoted by $p \mid q$ and $p \downarrow q$, respectively. (The operators \mid and \downarrow are called the **Sheffer stroke** and the **Peirce arrow** after H. M. Sheffer and C. S. Peirce, respectively.)

46. Construct a truth table for the logical operator *NAND*.
47. Show that $p \mid q$ is logically equivalent to $\neg(p \wedge q)$.
48. Construct a truth table for the logical operator *NOR*.
49. Show that $p \downarrow q$ is logically equivalent to $\neg(p \vee q)$.
50. In this exercise we will show that $\{\downarrow\}$ is a functionally complete collection of logical operators.

- a) Show that $p \downarrow p$ is logically equivalent to $\neg p$. b)

Show that $(p \downarrow q) \downarrow (p \downarrow q)$ is logically equivalent to $p \vee q$.

- c) Conclude from parts (a) and (b), and Exercise 49, that $\{\downarrow\}$ is a functionally complete collection of logical operators.

- *51. Find a compound proposition logically equivalent to $p \rightarrow q$ using only the logical operator \downarrow .
52. Show that $\{\downarrow\}$ is a functionally complete collection of logical operators.
53. Show that $p \mid q$ and $q \mid p$ are equivalent.
54. Show that $p \mid (q \mid r)$ and $(p \mid q) \mid r$ are not equivalent, so that the logical operator \mid is not associative. *

55. How many different truth tables of compound propositions are there that involve the propositional variables p and q ?

56. Show that if p , q , and r are compound propositions such that p and q are logically equivalent and q and r are logically equivalent, then p and r are logically equivalent.
57. The following sentence is taken from the specification of a telephone system: "If the directory database is opened, then the monitor is put in a closed state, if the system is not in its initial state." This specification is hard to under

stand because it involves two conditional statements. Find an equivalent, easier-to-understand specification that involves disjunctions and negations but not conditional statements.

58. How many of the disjunctions $p \vee \neg q$, $\neg p \vee q$, $q \vee r$, $q \vee \neg r$, and $\neg q \vee \neg r$ can be made simultaneously true by an assignment of truth values to p , q , and r ?
59. How many of the disjunctions $p \vee \neg q \vee s$, $\neg p \vee \neg r \vee s$, $\neg p \vee \neg r \vee \neg s$, $\neg p \vee q \vee \neg s$, $q \vee r \vee \neg s$, $q \vee \neg r \vee \neg s$, $\neg p \vee \neg q \vee \neg s$, $p \vee r \vee s$, and $p \vee r \vee \neg s$ can be made simultaneously true by an assignment of truth values to p , q , r , and s ?
60. Show that the negation of an unsatisfiable compound proposition is a tautology and the negation of a compound proposition that is a tautology is unsatisfiable.
61. Determine whether each of these compound propositions is satisfiable.
a) $(p \vee \neg q) \wedge (\neg p \vee q) \wedge (\neg p \vee \neg q)$
b) $(p \rightarrow q) \wedge (p \rightarrow \neg q) \wedge (\neg p \rightarrow q) \wedge (\neg p \rightarrow \neg q)$
c) $(p \leftrightarrow q) \wedge (\neg p \leftrightarrow q)$
62. Determine whether each of these compound propositions is satisfiable.
a) $(p \vee q \vee r) \wedge (p \vee \neg q \vee \neg s) \wedge (p \vee \neg r \vee \neg s) \wedge (\neg p \vee \neg q \vee \neg s) \wedge (p \vee q \vee \neg s)$
b) $(\neg p \vee \neg q \vee r) \wedge (\neg p \vee q \vee \neg s) \wedge (p \vee \neg q \vee \neg s) \wedge (\neg p \vee \neg r \vee \neg s) \wedge (p \vee q \vee \neg r) \wedge (p \vee \neg r \vee \neg s)$
c) $(p \vee q \vee r) \wedge (p \vee \neg q \vee \neg s) \wedge (q \vee \neg r \vee s) \wedge (\neg p \vee r \vee s) \wedge (\neg p \vee q \vee \neg s) \wedge (p \vee \neg q \vee \neg r) \wedge (\neg p \vee \neg q \vee s) \wedge (\neg p \vee \neg r \vee \neg s)$
63. Show how the solution of a given 4×4 Sudoku puzzle can be found by solving a satisfiability problem.
64. Construct a compound proposition that asserts that every cell of a 9×9 Sudoku puzzle contains at least one number.
65. Explain the steps in the construction of the compound proposition given in the text that asserts that every column of a 9×9 Sudoku puzzle contains every number.
*66. Explain the steps in the construction of the compound proposition given in the text that asserts that each of the nine 3×3 blocks of a 9×9 Sudoku puzzle contains every number.

statements in mathematics and in natural language. For example, suppose that we know that

“Every computer connected to the university network is functioning properly.”

1.4 Predicates and Quantifiers 37

No rules of propositional logic allow us to conclude the truth of the statement

“MATH3 is functioning properly,”

where MATH3 is one of the computers connected to the university network. Likewise, we cannot use the rules of propositional logic to conclude from the statement

“CS2 is under attack by an intruder,”

where CS2 is a computer on the university network, to conclude the truth of

“There is a computer on the university network that is under attack by an intruder.”

In this section we will introduce a more powerful type of logic called **predicate logic**. We will see how predicate logic can be used to express the meaning of a wide range of statements in mathematics and computer science in ways that permit us to reason and explore relationships between objects. To understand predicate logic, we first need to introduce the concept of a predicate. Afterward, we will introduce the notion of quantifiers, which enable us to reason with statements that assert that a certain property holds for all objects of a certain type and with statements that assert the existence of an object with a particular property.

Predicates

Statements involving variables, such as

“ $x > 3$,” “ $x = y + 3$,” “ $x + y = z$,”

and

“computer x is under attack by an intruder,”

and

“computer x is functioning properly,”

are often found in mathematical assertions, in computer programs, and in system specifications. These statements are neither true nor false when the values of the variables are not specified. In this section, we will discuss the ways that propositions can be produced from such statements.

The statement “ x is greater than 3” has two parts. The first part, the variable x , is the subject of the statement. The second part—the **predicate**, “is greater than 3”—refers to a property that the subject of the statement can have. We can denote the statement “ x is greater than 3” by $P(x)$, where P denotes the predicate “is greater than 3” and x is the variable. The statement $P(x)$ is also said to be the value of the **propositional function** P at x . Once a value has been assigned to the variable x , the statement $P(x)$ becomes a proposition and has a truth value. Consider Examples 1 and 2.

EXAMPLE 1 Let $P(x)$ denote the statement “ $x > 3$.” What are the truth values of $P(4)$ and $P(2)$?

Solution: We obtain the statement $P(4)$ by setting $x = 4$ in the statement “ $x > 3$.” Hence, $P(4)$, which is the statement “ $4 > 3$,” is true. However, $P(2)$, which is the statement “ $2 > 3$,” is

EXAMPLE 2 Let $A(x)$ denote the statement “Computer x is under attack by an intruder.” Suppose that of the computers on campus, only CS2 and MATH1 are currently under attack by intruders. What are truth values of $A(\text{CS1})$, $A(\text{CS2})$, and $A(\text{MATH1})$?

Solution: We obtain the statement $A(\text{CS1})$ by setting $x = \text{CS1}$ in the statement “Computer x is under attack by an intruder.” Because CS1 is not on the list of computers currently under attack, we conclude that $A(\text{CS1})$ is false. Similarly, because CS2 and MATH1 are on the list of computers under attack, we know that $A(\text{CS2})$ and $A(\text{MATH1})$ are true. .

We can also have statements that involve more than one variable. For instance, consider the statement “ $x = y + 3$.” We can denote this statement by $Q(x, y)$, where x and y are variables and Q is the predicate. When values are assigned to the variables x and y , the statement $Q(x, y)$ has a truth value.

EXAMPLE 3 Let $Q(x, y)$ denote the statement “ $x = y + 3$.” What are the truth values of the propositions $Q(1, 2)$ and $Q(3, 0)$?

Solution: To obtain $Q(1, 2)$, set $x = 1$ and $y = 2$ in the statement $Q(x, y)$. Hence, $Q(1, 2)$ is the statement “ $1 = 2 + 3$,” which is false. The statement $Q(3, 0)$ is the proposition “ $3 = 0 + 3$,” which is true. .



CHARLES SANDERS PEIRCE (1839–1914) Many consider Charles Peirce, born in Cambridge, Massachusetts, to be the most original and versatile American intellect. He made important contributions to an amazing number of disciplines, including mathematics, astronomy, chemistry, geodesy, metrology, engineering, psychology, philology, the history of science, and economics. Peirce was also an inventor, a lifelong student of medicine, a book reviewer, a dramatist and an actor, a short story writer, a phenomenologist, a logician, and a metaphysician. He is noted as the preeminent system-building philosopher competent and productive in logic, mathematics, and a wide range of sciences. He was encouraged by his father, Benjamin Peirce, a professor of mathematics and natural philosophy at Harvard, to pursue a career in science. Instead, he decided to study logic and scientific methodology. Peirce attended Harvard (1855–1859) and received a Harvard master of arts degree (1862) and an advanced degree in chemistry from the Lawrence Scientific School (1863).

In 1861, Peirce became an aide in the U.S. Coast Survey, with the goal of better understanding scientific methodology. His service for the Survey exempted him from military service during the Civil War. While working for the Survey, Peirce did astronomical and geodesic work. He made fundamental contributions to the design of pendulums and to map projections, applying new mathematical developments in the theory of elliptic functions. He was the first person to use the wavelength of light as a unit of measurement. Peirce rose to the position of Assistant for the Survey, a position he held until forced to resign in 1891 when he disagreed with the direction taken by the Survey’s new administration.

While making his living from work in the physical sciences, Peirce developed a hierarchy of sciences, with mathematics at the top rung, in which the methods of one science could be adapted for use by those sciences under it in the hierarchy. During this time, he also founded the American philosophical theory of pragmatism.

The only academic position Peirce ever held was lecturer in logic at Johns Hopkins University in Baltimore (1879–1884). His mathematical work during this time included contributions to logic, set theory, abstract algebra, and the philosophy of mathematics. His work is still relevant today, with recent applications of this work on logic to artificial intelligence. Peirce believed that the study of mathematics could develop the mind’s powers of imagination, abstraction, and generalization. His diverse activities after retiring from the Survey included writing for periodicals, contributing to scholarly dictionaries, translating scientific papers, guest lecturing, and textbook writing. Unfortunately, his income from these pursuits was insufficient to protect him and his second wife from abject poverty. He was supported in his later years by a fund created by his many admirers and administered by the philosopher William James, his lifelong friend. Although Peirce wrote and published voluminously in a vast range of subjects, he left more than 100,000 pages of unpublished manuscripts. Because of the difficulty of studying his unpublished writings, scholars have only recently started to understand some of his varied contributions. A group of people is devoted to making his work available over the Internet to bring a better appreciation of Peirce’s accomplishments to the world.

EXAMPLE 4 Let $A(c, n)$ denote the statement “Computer c is connected to network n ,” where c is a variable representing a computer and n is a variable representing a network. Suppose that the computer MATH1 is connected to network CAMPUS2, but not to network CAMPUS1. What are the values of $A(\text{MATH1}, \text{CAMPUS1})$ and $A(\text{MATH1}, \text{CAMPUS2})$?

Solution: Because MATH1 is not connected to the CAMPUS1 network, we see that $A(\text{MATH1}, \text{CAMPUS1})$ is false. However, because MATH1 is connected to the CAMPUS2 network, we see that $A(\text{MATH1}, \text{CAMPUS2})$ is true. .

Similarly, we can let $R(x, y, z)$ denote the statement “ $x + y = z$.” When values are assigned to the variables x , y , and z , this statement has a truth value.

EXAMPLE 5 What are the truth values of the propositions $R(1, 2, 3)$ and $R(0, 0, 1)$?

Solution: The proposition $R(1, 2, 3)$ is obtained by setting $x = 1$, $y = 2$, and $z = 3$ in the statement $R(x, y, z)$. We see that $R(1, 2, 3)$ is the statement “ $1 + 2 = 3$,” which is true. Also note that $R(0, 0, 1)$, which is the statement “ $0 + 0 = 1$,” is false. .

In general, a statement involving the n variables x_1, x_2, \dots, x_n can be denoted by

$$P(x_1, x_2, \dots, x_n).$$

A statement of the form $P(x_1, x_2, \dots, x_n)$ is the value of the **propositional function** P at the n -tuple (x_1, x_2, \dots, x_n) , and P is also called an **n -place predicate** or a **n -ary predicate**.

Propositional functions occur in computer programs, as Example 6 demonstrates.

EXAMPLE 6 Consider the statement

if $x > 0$ **then** $x := x + 1$.

When this statement is encountered in a program, the value of the variable x at that point in the execution of the program is inserted into $P(x)$, which is “ $x > 0$.” If $P(x)$ is true for this value of x , the assignment statement $x := x + 1$ is executed, so the value of x is increased by 1. If $P(x)$ is false for this value of x , the assignment statement is not executed, so the value of x is not changed. .

PRECONDITIONS AND POSTCONDITIONS Predicates are also used to establish the correctness of computer programs, that is, to show that computer programs always produce the desired output when given valid input. (Note that unless the correctness of a computer program is established, no amount of testing can show that it produces the desired output for all input values, unless every input value is tested.) The statements that describe valid input are known as **preconditions** and the conditions that the output should satisfy when the program has run are known as **postconditions**. As Example 7 illustrates, we use predicates to describe both preconditions and postconditions. We will study this process in greater detail in Section 5.5.

EXAMPLE 7 Consider the following program, designed to interchange the values of two variables x and y .

```
temp := x
x := y
y := temp
```

Find predicates that we can use as the precondition and the postcondition to verify the correctness of this program. Then explain how to use them to verify that for all valid input the program does what is intended.

Solution: For the precondition, we need to express that x and y have particular values before we run the program. So, for this precondition we can use the predicate $P(x, y)$, where $P(x, y)$ is the statement “ $x = a$ and $y = b$,” where a and b are the values of x and y before we run the program. Because we want to verify that the program swaps the values of x and y for all input values, for the postcondition we can use $Q(x, y)$, where $Q(x, y)$ is the statement “ $x = b$ and $y = a$.”

To verify that the program always does what it is supposed to do, suppose that the precondition $P(x, y)$ holds. That is, we suppose that the statement “ $x = a$ and $y = b$ ” is true. This means that $x = a$ and $y = b$. The first step of the program, $temp := x$, assigns the value of x to the variable $temp$, so after this step we know that $x = a$, $temp = a$, and $y = b$. After the second step of the program, $x := y$, we know that $x = b$, $temp = a$, and $y = b$. Finally, after the third step, we know that $x = b$, $temp = a$, and $y = a$. Consequently, after this program is run, the postcondition $Q(x, y)$ holds, that is, the statement “ $x = b$ and $y = a$ ” is true. .

Quantifiers

When the variables in a propositional function are assigned values, the resulting statement becomes a proposition with a certain truth value. However, there is another important way, called **quantification**, to create a proposition from a propositional function. Quantification expresses the extent to which a predicate is true over a range of elements. In English, the words *all*, *some*, *many*, *none*, and *few* are used in quantifications. We will focus on two types of quantification here: universal quantification, which tells us that a predicate is true for every element under consideration, and existential quantification, which tells us that there is one or more element under consideration for which the predicate is true. The area of logic that deals with predicates and quantifiers is called the **predicate calculus**.

THE UNIVERSAL QUANTIFIER Many mathematical statements assert that a property is true for all values of a variable in a particular domain, called the **domain of discourse** (or the **universe of discourse**), often just referred to as the **domain**. Such a statement is expressed using universal quantification. The universal quantification of $P(x)$ for a particular domain is the proposition that asserts that $P(x)$ is true for all values of x in this domain. Note that the domain specifies the possible values of the variable x . The meaning of the universal quantification of $P(x)$ changes when we change the domain. The domain must always be specified when a universal quantifier is used; without it, the universal quantification of a statement is not defined.

DEFINITION 1 The *universal quantification* of $P(x)$ is the statement

“ $P(x)$ for all values of x in the domain.”

The notation $\forall xP(x)$ denotes the universal quantification of $P(x)$. Here \forall is called the **universal quantifier**. We read $\forall xP(x)$ as “for all $xP(x)$ ” or “for every $xP(x)$.” An element for which $P(x)$ is false is called a **counterexample** of $\forall xP(x)$.

The meaning of the universal quantifier is summarized in the first row of Table 1. We illustrate the use of the universal quantifier in Examples 8–13.

TABLE 1 Quantifiers.		
Statement	When True?	When False?
$\forall xP(x)$ $\exists xP(x)$	$P(x)$ is true for every x . There is an x for which $P(x)$ is true.	There is an x for which $P(x)$ is false. $P(x)$ is false for every x .

EXAMPLE 8 Let $P(x)$ be the statement “ $x + 1 > x$.” What is the truth value of the quantification $\forall xP(x)$, where the domain consists of all real numbers?

Solution: Because $P(x)$ is true for all real numbers x , the quantification

$$\forall xP(x)$$

is true. .

Remark: Generally, an implicit assumption is made that all domains of discourse for quantifiers are nonempty. Note that if the domain is empty, then $\forall xP(x)$ is true for any propositional function $P(x)$ because there are no elements x in the domain for which $P(x)$ is false.

Remember that the truth value of $\forall xP(x)$ depends on the domain!

Besides “for all” and “for every,” universal quantification can be expressed in many other ways, including “all of,” “for each,” “given any,” “for arbitrary,” “for each,” and “for any.”

Remark: It is best to avoid using “for any x ” because it is often ambiguous as to whether “any” means “every” or “some.” In some cases, “any” is unambiguous, such as when it is used in negatives, for example, “there is not any

reason to avoid studying.”

A statement $\forall xP(x)$ is false, where $P(x)$ is a propositional function, if and only if $P(x)$ is not always true when x is in the domain. One way to show that $P(x)$ is not always true when x is in the domain is to find a counterexample to the statement $\forall xP(x)$. Note that a single counterexample is all we need to establish that $\forall xP(x)$ is false. Example 9 illustrates how counterexamples are used.

EXAMPLE 9 Let $Q(x)$ be the statement “ $x < 2$.” What is the truth value of the quantification $\forall xQ(x)$, where the domain consists of all real numbers?

Solution: $Q(x)$ is not true for every real number x , because, for instance, $Q(3)$ is false. That is, $x = 3$ is a counterexample for the statement $\forall xQ(x)$. Thus

$$\forall xQ(x)$$

is false. .

EXAMPLE 10 Suppose that $P(x)$ is “ $x^2 > 0$.” To show that the statement $\forall xP(x)$ is false where the universe of discourse consists of all integers, we give a counterexample. We see that $x = 0$ is a counterexample because $x^2 = 0$ when $x = 0$, so that x^2 is not greater than 0 when $x = 0$. .

Looking for counterexamples to universally quantified statements is an important activity in the study of mathematics, as we will see in subsequent sections of this book.

When all the elements in the domain can be listed—say, x_1, x_2, \dots, x_n —it follows that the universal quantification $\forall xP(x)$ is the same as the conjunction

$$P(x_1) \wedge P(x_2) \wedge \cdots \wedge P(x_n),$$

because this conjunction is true if and only if $P(x_1), P(x_2), \dots, P(x_n)$ are all true.

EXAMPLE 11 What is the truth value of $\forall xP(x)$, where $P(x)$ is the statement “ $x^2 < 10$ ” and the domain consists of the positive integers not exceeding 4?

Solution: The statement $\forall xP(x)$ is the same as the conjunction

$$P(1) \wedge P(2) \wedge P(3) \wedge P(4),$$

because the domain consists of the integers 1, 2, 3, and 4. Because $P(4)$, which is the statement “ $4^2 < 10$,” is false, it follows that $\forall xP(x)$ is false. .

EXAMPLE 12 What does the statement $\forall xN(x)$ mean if $N(x)$ is “Computer x is connected to the network” and the domain consists of all computers on campus?

Solution: The statement $\forall xN(x)$ means that for every computer x on campus, that computer x is connected to the network. This statement can be expressed in English as “Every computer on campus is connected to the network.” .

As we have pointed out, specifying the domain is mandatory when quantifiers are used. The truth value of a quantified statement often depends on which elements are in this domain, as Example 13 shows.

EXAMPLE 13 What is the truth value of $\forall x(x^2 \geq x)$ if the domain consists of all real numbers? What is the truth value of this statement if the domain consists of all integers?

Solution: The universal quantification $\forall x(x^2 \geq x)$, where the domain consists of all real numbers, is false. For example, $(\frac{1}{2})^2 \geq \frac{1}{2}$. Note that $x^2 \geq x$ if and only if $x^2 - x = x(x - 1) \geq 0$. Consequently, $x^2 \geq x$ if and only if $x \leq 0$ or $x \geq 1$. It follows that $\forall x(x^2 \geq x)$ is false if the domain consists of all real numbers (because the inequality is false for all real numbers x with $0 < x < 1$). However, if the domain consists of the integers, $\forall x(x^2 \geq x)$ is true, because there are no integers x with $0 < x < 1$. .

THE EXISTENTIAL QUANTIFIER Many mathematical statements assert that there is an element with a certain property. Such statements are expressed using existential quantification. With existential quantification, we form a proposition that is true if and only if $P(x)$ is true for at least one value of x in the domain.

DEFINITION 2 The *existential quantification* of $P(x)$ is the proposition

“There exists an element x in the domain such that $P(x)$.”

We use the notation $\exists xP(x)$ for the existential quantification of $P(x)$. Here \exists is called the *existential quantifier*.

A domain must always be specified when a statement $\exists xP(x)$ is used. Furthermore, the meaning of $\exists xP(x)$ changes when the domain changes. Without specifying the domain, the statement $\exists xP(x)$ has no meaning.

Besides the phrase “there exists,” we can also express existential quantification in many other ways, such as by using the words “for some,” “for at least one,” or “there is.” The existential quantification $\exists xP(x)$ is read as

“There is an x such that $P(x)$,”

“There is at least one x such that $P(x)$,”

or

“For some $xP(x)$.”

The meaning of the existential quantifier is summarized in the second row of Table 1. We illustrate the use of the existential quantifier in Examples 14–16.

EXAMPLE 14 Let $P(x)$ denote the statement “ $x > 3$.” What is the truth value of the quantification $\exists xP(x)$, where the domain consists of all real numbers?

Solution: Because “ $x > 3$ ” is sometimes true—for instance, when $x = 4$ —the existential quantification of $P(x)$, which is $\exists xP(x)$, is true. .

Observe that the statement $\exists xP(x)$ is false if and only if there is no element x in the domain for which $P(x)$ is true. That is, $\exists xP(x)$ is false if and only if $P(x)$ is false for every element of the domain. We illustrate this observation in Example 15.

EXAMPLE 15 Let $Q(x)$ denote the statement “ $x = x + 1$.” What is the truth value of the quantification $\exists xQ(x)$, where the domain consists of all real numbers?

Solution: Because $Q(x)$ is false for every real number x , the existential quantification of $Q(x)$, which is $\exists xQ(x)$, is false. .

Remember that the truth value of $\exists xP(x)$ depends

on the domain! **Remark:** Generally, an implicit assumption is made that all domains of discourse for quantifiers are nonempty. If the domain is empty, then $\exists xQ(x)$ is false whenever $Q(x)$ is a propositional function because when the domain is empty, there can be no element x in the domain for which $Q(x)$ is true.

When all elements in the domain can be listed—say, x_1, x_2, \dots, x_n —the existential quan

tification $\exists xP(x)$ is the same as the disjunction $P(x_1) \vee P(x_2) \vee \dots \vee P(x_n)$,

because this disjunction is true if and only if at least one of $P(x_1), P(x_2), \dots, P(x_n)$ is true.

EXAMPLE 16 What is the truth value of $\exists xP(x)$, where $P(x)$ is the statement “ $x^2 > 10$ ” and the universe of discourse consists of the positive integers not exceeding 4?

Solution: Because the domain is $\{1, 2, 3, 4\}$, the proposition $\exists xP(x)$ is the same as the disjunction

$$P(1) \vee P(2) \vee P(3) \vee P(4).$$

Because $P(4)$, which is the statement “ $4^2 > 10$,” is true, it follows that $\exists xP(x)$ is true. .

It is sometimes helpful to think in terms of looping and searching when determining the truth value of a quantification. Suppose that there are n objects in the domain for the variable x . To determine whether $\forall xP(x)$ is true, we can loop through all n values of x to see whether $P(x)$ is always true. If we encounter a value x for which $P(x)$ is false, then we have shown that $\forall xP(x)$ is false. Otherwise, $\forall xP(x)$ is true. To see whether $\exists xP(x)$ is true, we loop through the n values of x searching for a value for which $P(x)$ is true. If we find one, then $\exists xP(x)$ is true. If we never find such an x , then we have determined that $\exists xP(x)$ is false. (Note that this searching procedure does not apply if there are infinitely many values in the domain. However, it is still a useful way of thinking about the truth values of quantifications.)

THE UNIQUENESS QUANTIFIER We have now introduced universal and existential quantifiers. These are the most important quantifiers in mathematics and computer science. However, there is no limitation on the number of different quantifiers we can define, such as “there are exactly two,” “there are no more than three,” “there are at least 100,” and so on. Of these other quantifiers, the one that is most often seen is the **uniqueness quantifier**, denoted by $\exists!$ or \exists_1 . The notation $\exists! x P(x)$ [or $\exists_1 x P(x)$] states “There exists a unique x such that $P(x)$ is true.” (Other phrases for uniqueness quantification include “there is exactly one” and “there is one and only one.”) For instance, $\exists! x(x - 1 = 0)$, where the domain is the set of real numbers, states that there is a unique real number x such that $x - 1 = 0$. This is a true statement, as $x = 1$ is the unique real number such that $x - 1 = 0$. Observe that we can use quantifiers and propositional logic to express uniqueness (see Exercise 52 in Section 1.5), so the uniqueness quantifier can be avoided. Generally, it is best to stick with existential and universal quantifiers so that rules of inference for these quantifiers can be used.

Quantifiers with Restricted Domains

An abbreviated notation is often used to restrict the domain of a quantifier. In this notation, a condition a variable must satisfy is included after the quantifier. This is illustrated in Example 17. We will also describe other forms of this notation involving set membership in Section 2.1.

EXAMPLE 17 What do the statements $\forall x < 0 (x^2 > 0)$, $\forall y = 0 (y^3 = 0)$, and $\exists z > 0 (z^2 = 2)$ mean, where the domain in each case consists of the real numbers?

Solution: The statement $\forall x < 0 (x^2 > 0)$ states that for every real number x with $x < 0$, $x^2 > 0$. That is, it states “The square of a negative real number is positive.” This statement is the same as $\forall x(x < 0 \rightarrow x^2 > 0)$.

The statement $\forall y = 0 (y^3 = 0)$ states that for every real number y with $y = 0$, we have $y^3 = 0$. That is, it states “The cube of every nonzero real number is nonzero.” Note that this statement is equivalent to $\forall y(y = 0 \rightarrow y^3 = 0)$.

Finally, the statement $\exists z > 0 (z^2 = 2)$ states that there exists a real number z with $z > 0$ such that $z^2 = 2$. That is, it states “There is a positive square root of 2.” This statement is equivalent to $\exists z(z > 0 \wedge z^2 = 2)$.

Note that the restriction of a universal quantification is the same as the universal quantification of a conditional statement. For instance, $\forall x < 0 (x^2 > 0)$ is another way of expressing $\forall x(x < 0 \rightarrow x^2 > 0)$. On the other hand, the restriction of an existential quantification is the same as the existential quantification of a conjunction. For instance, $\exists z > 0 (z^2 = 2)$ is another way of expressing $\exists z(z > 0 \wedge z^2 = 2)$.

Precedence of Quantifiers

The quantifiers \forall and \exists have higher precedence than all logical operators from propositional calculus. For example, $\forall x P(x) \vee Q(x)$ is the disjunction of $\forall x P(x)$ and $Q(x)$. In other words, it means $(\forall x P(x)) \vee Q(x)$ rather than $\forall x(P(x) \vee Q(x))$.

Binding Variables

When a quantifier is used on the variable x , we say that this occurrence of the variable is **bound**. An occurrence of a variable that is not bound by a quantifier or set equal to a particular value is said to be **free**. All the variables that occur in a propositional function must be bound or set equal to a particular value to turn it into a proposition. This can be done using

The part of a logical expression to which a quantifier is applied is called the **scope** of this quantifier. Consequently, a variable is free if it is outside the scope of all quantifiers in the formula that specify this variable.

EXAMPLE 18 In the statement $\exists x(x + y = 1)$, the variable x is bound by the existential quantification $\exists x$, but the variable y is free because it is not bound by a quantifier and no value is assigned to this variable. This illustrates that in the statement $\exists x(x + y = 1)$, x is bound, but y is free.

In the statement $\exists x(P(x) \wedge Q(x)) \vee \forall xR(x)$, all variables are bound. The scope of the first quantifier, $\exists x$, is the expression $P(x) \wedge Q(x)$ because $\exists x$ is applied only to $P(x) \wedge Q(x)$, and not to the rest of the statement. Similarly, the scope of the second quantifier, $\forall x$, is the expression $R(x)$. That is, the existential quantifier binds the variable x in $P(x) \wedge Q(x)$ and the universal quantifier $\forall x$ binds the variable x in $R(x)$. Observe that we could have written our statement using two different variables x and y , as $\exists x(P(x) \wedge Q(x)) \vee \forall yR(y)$, because the scopes of the two quantifiers do not overlap. The reader should be aware that in common usage, the same letter is often used to represent variables bound by different quantifiers with scopes that do not overlap. .

Logical Equivalences Involving Quantifiers

In Section 1.3 we introduced the notion of logical equivalences of compound propositions. We can extend this notion to expressions involving predicates and quantifiers.

DEFINITION 3 Statements involving predicates and quantifiers are *logically equivalent* if and only if they have the same truth value no matter which predicates are substituted into these statements and which domain of discourse is used for the variables in these propositional functions. We use the notation $S \equiv T$ to indicate that two statements S and T involving predicates and quantifiers are logically equivalent.

Example 19 illustrates how to show that two statements involving predicates and quantifiers are logically equivalent.

EXAMPLE 19 Show that $\forall x(P(x) \wedge Q(x))$ and $\forall xP(x) \wedge \forall xQ(x)$ are logically equivalent (where the same domain is used throughout). This logical equivalence shows that we can distribute a universal quantifier over a conjunction. Furthermore, we can also distribute an existential quantifier over a disjunction. However, we cannot distribute a universal quantifier over a disjunction, nor can we distribute an existential quantifier over a conjunction. (See Exercises 50 and 51.)

Solution: To show that these statements are logically equivalent, we must show that they always take the same truth value, no matter what the predicates P and Q are, and no matter which domain of discourse is used. Suppose we have particular predicates P and Q , with a common domain. We can show that $\forall x(P(x) \wedge Q(x))$ and $\forall xP(x) \wedge \forall xQ(x)$ are logically equivalent by doing two things. First, we show that if $\forall x(P(x) \wedge Q(x))$ is true, then $\forall xP(x) \wedge \forall xQ(x)$ is true. Second, we show that if $\forall xP(x) \wedge \forall xQ(x)$ is true, then $\forall x(P(x) \wedge Q(x))$ is true.

So, suppose that $\forall x(P(x) \wedge Q(x))$ is true. This means that if a is in the domain, then $P(a) \wedge Q(a)$ is true. Hence, $P(a)$ is true and $Q(a)$ is true. Because $P(a)$ is true and $Q(a)$ is true for every element in the domain, we can conclude that $\forall xP(x)$ and $\forall xQ(x)$ are both true. This means that $\forall xP(x) \wedge \forall xQ(x)$ is true.

Next, suppose that $\forall xP(x) \wedge \forall xQ(x)$ is true. It follows that $\forall xP(x)$ is true and $\forall xQ(x)$ is true. Hence, if a is in the domain, then $P(a)$ is true and $Q(a)$ is true [because P

(x) and $Q(x)$ are both true for all elements in the domain, there is no conflict using the same value of a here].

It follows that for all a , $P(a) \wedge Q(a)$ is true. It follows that $\forall x(P(x) \wedge Q(x))$ is true. We can now conclude that

$$\forall x(P(x) \wedge Q(x)) \equiv \forall xP(x) \wedge \forall xQ(x).$$

Negating Quantified Expressions

We will often want to consider the negation of a quantified expression. For instance, consider the negation of the statement

“Every student in your class has taken a course in calculus.”

This statement is a universal quantification, namely,

$$\forall xP(x),$$

where $P(x)$ is the statement “ x has taken a course in calculus” and the domain consists of the students in your class. The negation of this statement is “It is not the case that every student in your class has taken a course in calculus.” This is equivalent to “There is a student in your class who has not taken a course in calculus.” And this is simply the existential quantification of the negation of the original propositional function, namely,

$$\exists x \neg P(x).$$

This example illustrates the following logical equivalence:

$$\neg \forall xP(x) \equiv \exists x \neg P(x).$$

To show that $\neg \forall xP(x)$ and $\exists x \neg P(x)$ are logically equivalent no matter what the propositional function $P(x)$ is and what the domain is, first note that $\neg \forall xP(x)$ is true if and only if $\forall xP(x)$ is false. Next, note that $\forall xP(x)$ is false if and only if there is an element x in the domain for which $P(x)$ is false. This holds if and only if there is an element x in the domain for which $\neg P(x)$ is true. Finally, note that there is an element x in the domain for which $\neg P(x)$ is true if and only if $\exists x \neg P(x)$ is true. Putting these steps together, we can conclude that $\neg \forall xP(x)$ is true if and only if $\exists x \neg P(x)$ is true. It follows that $\neg \forall xP(x)$ and $\exists x \neg P(x)$ are logically equivalent.

Suppose we wish to negate an existential quantification. For instance, consider the proposition “There is a student in this class who has taken a course in calculus.” This is the existential quantification

$$\exists xQ(x),$$

where $Q(x)$ is the statement “ x has taken a course in calculus.” The negation of this statement is the proposition “It is not the case that there is a student in this class who has taken a course in calculus.” This is equivalent to “Every student in this class has not taken calculus,” which is just the universal quantification of the negation of the original propositional function, or, phrased in the language of quantifiers,

$$\forall x \neg Q(x).$$

This example illustrates the equivalence

$$\neg \exists xQ(x) \equiv \forall x \neg Q(x).$$

To show that $\neg \exists xQ(x)$ and $\forall x \neg Q(x)$ are logically equivalent no matter what $Q(x)$ is and

what the domain is, first note that $\neg \exists x Q(x)$ is true if and only if $\exists x Q(x)$ is false. This is true if and

TABLE 2 De Morgan's Laws for Quantifiers.			
<i>Negation</i>	<i>Equivalent Statement</i>	<i>When Is Negation True?</i>	<i>When False?</i>
$\neg \exists x P(x)$	$\forall x \neg P(x)$	For every x , $P(x)$ is false.	There is an x for which $P(x)$ is true.
$\neg \forall x P(x)$	$\exists x \neg P(x)$	There is an x for which $P(x)$ is false.	$P(x)$ is true for every x .

only if no x exists in the domain for which $Q(x)$ is true. Next, note that no x exists in the domain for which $Q(x)$ is true if and only if $Q(x)$ is false for every x in the domain. Finally, note that $Q(x)$ is false for every x in the domain if and only if $\neg Q(x)$ is true for all x in the domain, which holds if and only if $\forall x \neg Q(x)$ is true. Putting these steps together, we see that $\neg \exists x Q(x)$ is true if and only if $\forall x \neg Q(x)$ is true. We conclude that $\neg \exists x Q(x)$ and $\forall x \neg Q(x)$ are logically equivalent.

The rules for negations for quantifiers are called **De Morgan's laws for quantifiers**. These rules are summarized in Table 2.

Remark: When the domain of a predicate $P(x)$ consists of n elements, where n is a positive integer greater than one, the rules for negating quantified statements are exactly the same as De Morgan's laws discussed in Section 1.3. This is why these rules are called De Morgan's laws for quantifiers. When the domain has n elements x_1, x_2, \dots, x_n , it follows that $\neg \forall x P(x)$ is the same as $\neg(P(x_1) \wedge P(x_2) \wedge \dots \wedge P(x_n))$, which is equivalent to $\neg P(x_1) \vee \neg P(x_2) \vee \dots \vee \neg P(x_n)$ by De Morgan's laws, and this is the same as $\exists x \neg P(x)$. Similarly, $\neg \exists x P(x)$ is the same as $\neg(P(x_1) \vee P(x_2) \vee \dots \vee P(x_n))$, which by De Morgan's laws is equivalent to $\neg P(x_1) \wedge \neg P(x_2) \wedge \dots \wedge \neg P(x_n)$, and this is the same as $\forall x \neg P(x)$.

We illustrate the negation of quantified statements in Examples 20 and 21.

EXAMPLE 20 What are the negations of the statements “There is an honest politician” and “All Americans eat cheeseburgers”?

Solution: Let $H(x)$ denote “ x is honest.” Then the statement “There is an honest politician” is represented by $\exists x H(x)$, where the domain consists of all politicians. The negation of this statement is $\neg \exists x H(x)$, which is equivalent to $\forall x \neg H(x)$. This negation can be expressed as “Every politician is dishonest.” (Note: In English, the statement “All politicians are not honest” is ambiguous. In common usage, this statement often means “Not all politicians are honest.” Consequently, we do not use this statement to express this negation.)

Let $C(x)$ denote “ x eats cheeseburgers.” Then the statement “All Americans eat cheeseburgers” is represented by $\forall x C(x)$, where the domain consists of all Americans. The negation of this statement is $\neg \forall x C(x)$, which is equivalent to $\exists x \neg C(x)$. This negation can be expressed in several different ways, including “Some American does not eat cheeseburgers” and “There is an American who does not eat cheeseburgers.”

EXAMPLE 21 What are the negations of the statements $\forall x (x^2 > x)$ and $\exists x (x^2 = 2)$?

Solution: The negation of $\forall x (x^2 > x)$ is the statement $\neg \forall x (x^2 > x)$, which is equivalent to $\exists x \neg (x^2 > x)$. This can be rewritten as $\exists x (x^2 \leq x)$. The negation of $\exists x (x^2 = 2)$ is the statement $\neg \exists x (x^2 = 2)$, which is equivalent to $\forall x \neg (x^2 = 2)$. This can be rewritten as $\forall x (x^2 \neq 2)$. The truth values of these statements depend on the domain.

EXAMPLE 22 Show that $\neg \forall x(P(x) \rightarrow Q(x))$ and $\exists x(P(x) \wedge \neg Q(x))$ are logically equivalent.

Solution: By De Morgan's law for universal quantifiers, we know that $\neg \forall x(P(x) \rightarrow Q(x))$ and $\exists x(\neg(P(x) \rightarrow Q(x)))$ are logically equivalent. By the fifth logical equivalence in Table 7 in Section 1.3, we know that $\neg(P(x) \rightarrow Q(x))$ and $P(x) \wedge \neg Q(x)$ are logically equivalent for every x . Because we can substitute one logically equivalent expression for another in a logical equivalence, it follows that $\neg \forall x(P(x) \rightarrow Q(x))$ and $\exists x(P(x) \wedge \neg Q(x))$ are logically equivalent. .

Translating from English into Logical Expressions

Translating sentences in English (or other natural languages) into logical expressions is a crucial task in mathematics, logic programming, artificial intelligence, software engineering, and many other disciplines. We began studying this topic in Section 1.1, where we used propositions to express sentences in logical expressions. In that discussion, we purposely avoided sentences whose translations required predicates and quantifiers. Translating from English to logical expressions becomes even more complex when quantifiers are needed. Furthermore, there can be many ways to translate a particular sentence. (As a consequence, there is no “cookbook” approach that can be followed step by step.) We will use some examples to illustrate how to translate sentences from English into logical expressions. The goal in this translation is to produce simple and useful logical expressions. In this section, we restrict ourselves to sentences that can be translated into logical expressions using a single quantifier; in the next section, we will look at more complicated sentences that require multiple quantifiers.

EXAMPLE 23 Express the statement “Every student in this class has studied calculus” using predicates and quantifiers.

Solution: First, we rewrite the statement so that we can clearly identify the appropriate quantifiers to use. Doing so, we obtain:

“For every student in this class, that student has studied calculus.”

Next, we introduce a variable x so that our statement becomes

“For every student x in this class, x has studied calculus.”

Continuing, we introduce $C(x)$, which is the statement “ x has studied calculus.” Consequently, if the domain for x consists of the students in the class, we can translate our statement as $\forall x C(x)$. However, there are other correct approaches; different domains of discourse and other predicates can be used. The approach we select depends on the subsequent reasoning we want to carry out. For example, we may be interested in a wider group of people than only those in this class. If we change the domain to consist of all people, we will need to express our statement as

“For every person x , if person x is a student in this class then x has studied calculus.”

If $S(x)$ represents the statement that person x is in this class, we see that our statement can be expressed as $\forall x(S(x) \rightarrow C(x))$. [Caution! Our statement *cannot* be expressed as $\forall x(S(x) \wedge C(x))$ because this statement says that all people are students in this class and have studied calculus!]

Finally, when we are interested in the background of people in subjects besides calculus, we may prefer to use the two-variable quantifier $Q(x, y)$ for the statement “student x has studied subject y .” Then we would replace $C(x)$ by $Q(x, \text{calculus})$ in both approaches to

In Example 23 we displayed different approaches for expressing the same statement using predicates and quantifiers. However, we should always adopt the simplest approach that is adequate for use in subsequent reasoning.

EXAMPLE 24 Express the statements “Some student in this class has visited Mexico” and “Every student in this class has visited either Canada or Mexico” using predicates and quantifiers.

Solution: The statement “Some student in this class has visited Mexico” means that

“There is a student in this class with the property that the student has visited Mexico.” We can introduce a variable x , so that our statement becomes

“There is a student x in this class having the property that x has visited Mexico.”

We introduce $M(x)$, which is the statement “ x has visited Mexico.” If the domain for x consists of the students in this class, we can translate this first statement as $\exists xM(x)$.

However, if we are interested in people other than those in this class, we look at the statement a little differently. Our statement can be expressed as

“There is a person x having the properties that x is a student in this class and x has visited Mexico.”

In this case, the domain for the variable x consists of all people. We introduce $S(x)$ to represent “ x is a student in this class.” Our solution becomes $\exists x(S(x) \wedge M(x))$ because the statement is that there is a person x who is a student in this class and who has visited Mexico. [Caution! Our statement cannot be expressed as $\exists x(S(x) \rightarrow M(x))$, which is true when there is someone not in the class because, in that case, for such a person x , $S(x) \rightarrow M(x)$ becomes either $F \rightarrow T$ or $F \rightarrow F$, both of which are true.]

Similarly, the second statement can be expressed as

“For every x in this class, x has the property that x has visited Mexico or x has visited Canada.”

(Note that we are assuming the inclusive, rather than the exclusive, or here.) We let $C(x)$ be “ x has visited Canada.” Following our earlier reasoning, we see that if the domain for x consists of the students in this class, this second statement can be expressed as $\forall x(C(x) \vee M(x))$. However, if the domain for x consists of all people, our statement can be expressed as

“For every person x , if x is a student in this class, then x has visited Mexico or x has visited Canada.”

In this case, the statement can be expressed as $\forall x(S(x) \rightarrow (C(x) \vee M(x)))$.

Instead of using $M(x)$ and $C(x)$ to represent that x has visited Mexico and x has visited Canada, respectively, we could use a two-place predicate $V(x, y)$ to represent “ x has visited country y .” In this case, $V(x, \text{Mexico})$ and $V(x, \text{Canada})$ would have the same meaning as $M(x)$ and $C(x)$ and could replace them in our answers. If we are working with many statements that involve people visiting different countries, we might prefer to use this two-variable approach. Otherwise, for simplicity, we would stick with the one-variable predicates $M(x)$ and $C(x)$.

Using Quantifiers in System Specifications

In Section 1.2 we used propositions to represent system specifications. However, many system specifications involve predicates and quantifications. This is illustrated in Example 25.

EXAMPLE 25 Use predicates and quantifiers to express the system specifications “Every mail message larger than one megabyte will be compressed” and “If a user is active, at least one network link will be available.”

Solution: Let $S(m, y)$ be “Mail message m is larger than y megabytes,” where the variable x has the domain of all mail messages and the variable y is a positive real number, and let $C(m)$ denote “Mail message m will be compressed.” Then the specification “Every mail message larger than one megabyte will be compressed” can be represented as $\forall m(S(m, 1) \rightarrow C(m))$. Remember the rules of

precedence for quantifiers and logical connectives!

Let $A(u)$ represent “User u is active,” where the variable u has the domain of all users, let $S(n, x)$ denote “Network link n is in state x ,” where n has the domain of all network links and x has the domain of all possible states for a network link. Then the specification “If a user is active, at least one network link will be available” can be represented by $\exists u A(u) \rightarrow \exists n S(n, \text{available})$.

Lewis Carroll (really C. L. Dodgson writing under a pseudonym), the author of *Alice in Wonderland*, is also the author of several works on symbolic logic. His books contain many examples of reasoning using quantifiers. Examples 26 and 27 come from his book *Symbolic Logic*; other examples from that book are given in the exercises at the end of this section. These examples illustrate how quantifiers are used to express various types of statements.

Examples from Lewis Carroll

EXAMPLE 26 Consider these statements. The first two are called *premises* and the third is called the *conclusion*. The entire set is called an *argument*.

“All lions are fierce.”

“Some lions do not drink coffee.”

“Some fierce creatures do not drink coffee.”

(In Section 1.6 we will discuss the issue of determining whether the conclusion is a valid consequence of the premises. In this example, it is.) Let $P(x)$, $Q(x)$, and $R(x)$ be the statements “ x is a lion,” “ x is fierce,” and “ x drinks coffee,” respectively. Assuming that the domain consists of all creatures, express the statements in the argument using quantifiers and $P(x)$, $Q(x)$, and $R(x)$.



CHARLES LUTWIDGE DODGSON (1832–1898) We know Charles Dodgson as Lewis Carroll—the pseudonym he used in his literary works. Dodgson, the son of a clergyman, was the third of 11 children, all of whom stuttered. He was uncomfortable in the company of adults and is said to have spoken without stuttering only to young girls, many of whom he entertained, corresponded with, and photographed (sometimes in poses that today would be considered inappropriate). Although attracted to young girls, he was extremely puritanical and religious. His friendship with the three young daughters of Dean Liddell led to his writing *Alice in Wonderland*, which brought him money and fame.

Dodgson graduated from Oxford in 1854 and obtained his master of arts degree in 1857. He was appointed lecturer in mathematics at Christ Church College, Oxford, in 1855. He was ordained in the Church of England in 1861 but never practiced his ministry. His writings published under this real name include articles and books on geometry, determinants, and the mathematics of tournaments and elections. (He also used the pseudonym Lewis Carroll for his many works on recreational logic.)

Solution: We can express these statements as:

$$\begin{aligned}\forall x(P(x) \rightarrow Q(x)). \\ \exists x(P(x) \wedge \neg R(x)). \\ \exists x(Q(x) \wedge \neg R(x)).\end{aligned}$$

Notice that the second statement cannot be written as $\exists x(P(x) \rightarrow \neg R(x))$. The reason is that $P(x) \rightarrow \neg R(x)$ is true whenever x is not a lion, so that $\exists x(P(x) \rightarrow \neg R(x))$ is true as long as there is at least one creature that is not a lion, even if every lion drinks coffee. Similarly, the third statement cannot be written as

$$\exists x(Q(x) \rightarrow \neg R(x)).$$

EXAMPLE 27 Consider these statements, of which the first three are premises and the fourth is a valid conclusion.

“All hummingbirds are richly colored.”
 “No large birds live on honey.”
 “Birds that do not live on honey are dull in color.”
 “Hummingbirds are small.”

Let $P(x)$, $Q(x)$, $R(x)$, and $S(x)$ be the statements “ x is a hummingbird,” “ x is large,” “ x lives on honey,” and “ x is richly colored,” respectively. Assuming that the domain consists of all birds, express the statements in the argument using quantifiers and $P(x)$, $Q(x)$, $R(x)$, and $S(x)$.

Solution: We can express the statements in the argument as

$$\begin{aligned}\forall x(P(x) \rightarrow S(x)). \\ \neg \exists x(Q(x) \wedge R(x)). \\ \forall x(\neg R(x) \rightarrow \neg S(x)). \\ \forall x(P(x) \rightarrow \neg Q(x)).\end{aligned}$$

(Note we have assumed that “small” is the same as “not large” and that “dull in color” is the same as “not richly colored.” To show that the fourth statement is a valid conclusion of the first three, we need to use rules of inference that will be discussed in Section 1.6.)

Logic Programming

An important type of programming language is designed to reason using the rules of predicate logic. Prolog (from *Programming in Logic*), developed in the 1970s by computer scientists working in the area of artificial intelligence, is an example of such a language. Prolog programs include a set of declarations consisting of two types of statements, **Prolog facts** and **Prolog rules**. Prolog facts define predicates by specifying the elements that satisfy these predicates. Prolog rules are used to define new predicates using those already defined by Prolog facts. Example 28 illustrates these notions.

EXAMPLE 28 Consider a Prolog program given facts telling it the instructor of each class and in which classes students are enrolled. The program uses these facts to answer queries concerning the professors who teach particular students. Such a program could use the predicates $instructor(p, c)$ and

$enrolled(s, c)$ to represent that professor p is the instructor of course c and that student s is enrolled in course c , respectively. For example, the Prolog facts in such a program might

include:

```
instructor(chan,math273)
instructor(patel,ee222)
instructor(grossman,cs301)
enrolled(kevin,math273)
enrolled(juana,ee222)
enrolled(juana,cs301)
enrolled(kiko,math273)
enrolled(kiko,cs301)
```

(Lowercase letters have been used for entries because Prolog considers names beginning with an uppercase letter to be variables.)

A new predicate *teaches*(*p*, *s*), representing that professor *p* teaches student *s*, can be defined using the Prolog rule

```
teaches(P,S) :- instructor(P,C) , enrolled(S,C)
```

which means that *teaches*(*p*, *s*) is true if there exists a class *c* such that professor *p* is the instructor of class *c* and student *s* is enrolled in class *c*. (Note that a comma is used to represent a conjunction of predicates in Prolog. Similarly, a semicolon is used to represent a disjunction of predicates.)

Prolog answers queries using the facts and rules it is given. For example, using the facts and rules listed, the query

```
?enrolled(kevin,math273)
```

produces the response

```
yes
```

because the fact *enrolled*(kevin, math273) was provided as input. The query

```
?enrolled(X,math273)
```

produces the response

```
kevin
kiko
```

To produce this response, Prolog determines all possible values of *X* for which *enrolled*(*X*, math273) has been included as a Prolog fact. Similarly, to find all the professors who are instructors in classes being taken by Juana, we use the query

```
?teaches(X,juana)
```

This query returns

```
patel
grossman
```

Exercises

1. Let $P(x)$ denote the statement “ $x \leq 4$.” What are these truth values?
a) $P(0)$ b) $P(4)$ c) $P(6)$
2. Let $P(x)$ be the statement

“the word x contains the letter a .” What are these truth values?

- a) $P(\text{orange})$ b) $P(\text{lemon})$
- c) $P(\text{true})$ d) $P(\text{false})$

3. Let $Q(x, y)$ denote the statement “ x is the capital of y .”

What are these truth values?

- a) $Q(\text{Denver, Colorado})$
 - b) $Q(\text{Detroit, Michigan})$
 - c) $Q(\text{Massachusetts, Boston})$
 - d) $Q(\text{New York, New York})$
4. State the value of x after the statement **if** $P(x)$ **then** $x := 1$ is executed, where $P(x)$ is the statement “ $x > 1$,” if the value of x when this statement is reached is
- a) $x = 0$. b) $x = 1$.
 - c) $x = 2$.
5. Let $P(x)$ be the statement “ x spends more than five hours every weekday in class,” where the domain for x consists of all students. Express each of these quantifications in English.
- a) $\exists x P(x)$ b) $\forall x P(x)$
 - c) $\exists x \neg P(x)$ d) $\forall x \neg P(x)$
6. Let $N(x)$ be the statement “ x has visited North Dakota,” where the domain consists of the students in your school. Express each of these quantifications in English.
- a) $\exists x N(x)$ b) $\forall x N(x)$ c) $\neg \exists x N(x)$ d) $\exists x \neg N(x)$ e) $\neg \forall x N(x)$ f) $\forall x \neg N(x)$
7. Translate these statements into English, where $C(x)$ is “ x is a comedian” and $F(x)$ is “ x is funny” and the domain consists of all people.
- a) $\forall x (C(x) \rightarrow F(x))$ b) $\forall x (C(x) \wedge F(x))$ c) $\exists x (C(x) \rightarrow F(x))$ d) $\exists x (C(x) \wedge F(x))$
8. Translate these statements into English, where $R(x)$ is “ x is a rabbit” and $H(x)$ is “ x hops” and the domain consists of all animals.
- a) $\forall x (R(x) \rightarrow H(x))$ b) $\forall x (R(x) \wedge H(x))$ c) $\exists x (R(x) \rightarrow H(x))$ d) $\exists x (R(x) \wedge H(x))$
9. Let $P(x)$ be the statement “ x can speak Russian” and let $Q(x)$ be the statement “ x knows the computer language C++.” Express each of these sentences in terms of $P(x)$, $Q(x)$, quantifiers, and logical connectives. The domain for quantifiers consists of all students at your school.
- a) There is a student at your school who can speak Russian and who knows C++.
 - b) There is a student at your school who can speak Russian but who doesn’t know C++.
 - c) Every student at your school either can speak Russian or knows C++.
 - d) No student at your school can speak Russian or knows C++.
10. Let $C(x)$ be the statement “ x has a cat,” let $D(x)$ be the statement “ x has a dog,” and let $F(x)$ be the statement “ x has a ferret.” Express each of these statements in terms of $C(x)$, $D(x)$, $F(x)$, quantifiers, and logical connectives. Let the domain consist of all students in your class.
- a) A student in your class has a cat, a dog, and a ferret.
 - b) All students in your class have a cat, a dog, or a ferret.
 - c) Some student in your class has a cat and a ferret, but not a dog.
 - d) No student in your class has a cat, a dog, and a ferret.
 - e) For each of the three animals, cats, dogs, and ferrets, there is a student in your class who has this animal as a pet.
11. Let $P(x)$ be the statement “ $x = x^2$.” If the domain consists of the integers, what are these truth values? a) $P(0)$ b) $P(1)$ c) $P(2)$ d) $P(-1)$ e) $\exists x P(x)$ f) $\forall x P(x)$
12. Let $Q(x)$ be the statement “ $x + 1 > 2x$.” If the domain consists of all integers, what are these truth values? a) $Q(0)$ b) $Q(-1)$ c) $Q(1)$ d) $\exists x Q(x)$ e) $\forall x Q(x)$ f) $\exists x \neg Q(x)$ g) $\forall x \neg Q(x)$
13. Determine the truth value of each of these statements if the domain consists of all integers.
- a) $\forall n (n + 1 > n)$ b) $\exists n (2n = 3n)$
 - c) $\exists n (n = -n)$ d) $\forall n (3n \leq 4n)$
14. Determine the truth value of each of these statements if the domain consists of all real numbers.
- a) $\exists x (x^3 = -1)$ b) $\exists x (x^4 < x^2)$
 - c) $\forall x ((-x)^2 = x^2)$ d) $\forall x (2x > x)$
15. Determine the truth value of each of these statements if the domain for all variables consists of all integers.
- a) $\forall n (n^2 \geq 0)$ b) $\exists n (n^2 = 2)$
 - c) $\forall n (n^2 \geq n)$ d) $\exists n (n^2 < 0)$
16. Determine the truth value of each of these statements if the domain of each variable consists of all real numbers.
- a) $\exists x (x^2 = 2)$ b) $\exists x (x^2 = -1)$ c) $\forall x (x^2 + 2 \geq 1)$ d) $\forall x (x^2 = x)$
17. Suppose that the domain of the propositional function $P(x)$ consists of the integers 0, 1, 2, 3, and 4. Write out each of these propositions using disjunctions, conjunctions, and negations.
- a) $\exists x P(x)$ b) $\forall x P(x)$ c) $\exists x \neg P(x)$ d) $\forall x \neg P(x)$ e) $\neg \exists x P(x)$ f) $\neg \forall x P(x)$
18. Suppose that the domain of the propositional function $P(x)$ consists of the integers $-2, -1, 0, 1$, and 2 . Write out each of these propositions using disjunctions, conjunctions, and negations.
- a) $\exists x P(x)$ b) $\forall x P(x)$ c) $\exists x \neg P(x)$ d) $\forall x \neg P(x)$ e) $\neg \exists x P(x)$ f) $\neg \forall x P(x)$
- 54 1 / The Foundations: Logic and Proofs
19. Suppose that the domain of the propositional function $P(x)$ consists of the integers 1, 2, 3, 4, and 5. Express these statements without using quantifiers, instead using only negations, disjunctions, and conjunctions.
- a) $\exists x P(x)$ b) $\forall x P(x)$
 - c) $\neg \exists x P(x)$ d) $\neg \forall x P(x)$
 - e) $\forall x ((x = 3) \rightarrow P(x)) \vee \exists x \neg P(x)$
20. Suppose that the domain of the propositional function $P(x)$ consists of $-5, -3, -1, 1, 3$, and 5 . Express these statements without using quantifiers, instead using only negations, disjunctions, and conjunctions.
- a) $\exists x P(x)$ b) $\forall x P(x)$
 - c) $\forall x ((x = 1) \rightarrow P(x))$
 - d) $\exists x ((x \geq 0) \wedge P(x))$
 - e) $\exists x (\neg P(x)) \wedge \forall x ((x < 0) \rightarrow P(x))$
21. For each of these statements find a domain for which the statement is true and a domain for which the statement is false.
- a) Everyone is studying discrete mathematics.
 - b) Everyone is older than 21 years.
 - c) Every two people have the same mother.
 - d) No two different people have the same grandmother.
22. For each of these statements find a domain for which the statement is true and a domain for which the statement

is false.

- a) Everyone speaks Hindi.
- b) There is someone older than 21 years.
- c) Every two people have the same first name. d)

Someone knows more than two other people. **23.** Translate in two ways each of these statements into logical expressions using predicates, quantifiers, and logical connectives. First, let the domain consist of the students in your class and second, let it consist of all people. a)

Someone in your class can speak Hindi.

- b) Everyone in your class is friendly.
- c) There is a person in your class who was not born in California.
- d) A student in your class has been in a movie. e) No student in your class has taken a course in logic programming.

24. Translate in two ways each of these statements into logical expressions using predicates, quantifiers, and logical connectives. First, let the domain consist of the students in your class and second, let it consist of all people.

- a) Everyone in your class has a cellular phone. b) Somebody in your class has seen a foreign movie. c) There is a person in your class who cannot swim. d) All students in your class can solve quadratic equations.
- e) Some student in your class does not want to be rich.

25. Translate each of these statements into logical expressions using predicates, quantifiers, and logical connectives.

- a) No one is perfect.
- b) Not everyone is perfect.
- c) All your friends are perfect.
- d) At least one of your friends is perfect.

- e) Everyone is your friend and is perfect.
- f) Not everybody is your friend or someone is not perfect.

26. Translate each of these statements into logical expressions in three different ways by varying the domain and by using predicates with one and with two variables. a) Someone in your school has visited Uzbekistan. b) Everyone in your class has studied calculus and C++.

c) No one in your school owns both a bicycle and a motorcycle.

d) There is a person in your school who is not happy.

e) Everyone in your school was born in the twentieth century.

27. Translate each of these statements into logical expressions in three different ways by varying the domain and by using predicates with one and with two variables. a) A student in your school has lived in Vietnam. b) There is a student in your school who cannot speak Hindi.

- c) A student in your school knows Java, Prolog, and C++.
- d) Everyone in your class enjoys Thai food.
- e) Someone in your class does not play hockey. **28.**

Translate each of these statements into logical expressions using predicates, quantifiers, and logical connectives.

- a) Something is not in the correct place.

- b) All tools are in the correct place and are in excellent condition.
- c) Everything is in the correct place and in excellent condition.
- d) Nothing is in the correct place and is in excellent condition.
- e) One of your tools is not in the correct place, but it is in excellent condition.

29. Express each of these statements using logical operators, predicates, and quantifiers.

- a) Some propositions are tautologies.
- b) The negation of a contradiction is a tautology. c) The disjunction of two contingencies can be a tautology.
- d) The conjunction of two tautologies is a tautology.

30. Suppose the domain of the propositional function $P(x, y)$ consists of pairs x and y , where x is 1, 2, or 3 and y is 1, 2, or 3. Write out these propositions using disjunctions and conjunctions.

- a) $\exists x P(x, 3)$ b) $\forall y P(1, y)$
- c) $\exists y \neg P(2, y)$ d) $\forall x \neg P(x, 2)$

31. Suppose that the domain of $Q(x, y, z)$ consists of triples x, y, z , where $x = 0, 1, \text{ or } 2$, $y = 0 \text{ or } 1$, and $z = 0 \text{ or } 1$. Write out these propositions using disjunctions and conjunctions.

- a) $\forall y Q(0, y, 0)$ b) $\exists x Q(x, 1, 1)$
- c) $\exists z \neg Q(0, 0, z)$ d) $\exists x \neg Q(x, 0, 1)$

32. Express each of these statements using quantifiers. Then form the negation of the statement so that no negation is to the left of a quantifier. Next, express the negation in simple English. (Do not simply use the phrase "It is not the case that.")

- a) All dogs have fleas.
- b) There is a horse that can add.
- c) Every koala can climb.
- d) No monkey can speak French.
- e) There exists a pig that can swim and catch fish. **33.**

Express each of these statements using quantifiers. Then form the negation of the statement, so that no negation is to the left of a quantifier. Next, express the negation in simple English. (Do not simply use the phrase "It is not the case that.")

- a) Some old dogs can learn new tricks.
- b) No rabbit knows calculus.
- c) Every bird can fly.
- d) There is no dog that can talk.
- e) There is no one in this class who knows French and Russian.

34. Express the negation of these propositions using quantifiers, and then express the negation in English. a) Some drivers do not obey the speed limit.

- b) All Swedish movies are serious.
- c) No one can keep a secret.
- d) There is someone in this class who does not have a good attitude.

35. Find a counterexample, if possible, to these universally quantified statements, where the domain for all variables consists of all integers.

- a) $\forall x(x^2 \geq x)$
 b) $\forall x(x > 0 \vee x < 0)$
 c) $\forall x(x = 1)$
36. Find a counterexample, if possible, to these universally quantified statements, where the domain for all variables consists of all real numbers.
 a) $\forall x(x^2 = x)$ b) $\forall x(x^2 = 2)$
 c) $\forall x(|x| > 0)$
37. Express each of these statements using predicates and quantifiers.
 a) A passenger on an airline qualifies as an elite flyer if the passenger flies more than 25,000 miles in a year or takes more than 25 flights during that year.
 b) A man qualifies for the marathon if his best previous time is less than 3 hours and a woman qualifies for the marathon if her best previous time is less than 3.5 hours.
 c) A student must take at least 60 course hours, or at least 45 course hours and write a master's thesis, and receive a grade no lower than a B in all required courses, to receive a master's degree.
 d) There is a student who has taken more than 21 credit hours in a semester and received all A's.

1.4 Predicates and Quantifiers 55

Exercises 38–42 deal with the translation between system specification and logical expressions involving quantifiers.

38. Translate these system specifications into English where the predicate $S(x, y)$ is “ x is in state y ” and where the domain for x and y consists of all systems and all possible states, respectively.
 a) $\exists xS(x, \text{open})$
 b) $\forall x(S(x, \text{malfunctioning}) \vee S(x, \text{diagnostic}))$ c) $\exists xS(x, \text{open}) \vee \exists xS(x, \text{diagnostic})$
 d) $\exists x\neg S(x, \text{available})$
 e) $\forall x\neg S(x, \text{working})$
39. Translate these specifications into English where $F(p)$ is “Printer p is out of service,” $B(p)$ is “Printer p is busy,” $L(j)$ is “Print job j is lost,” and $Q(j)$ is “Print job j is queued.”
 a) $\exists p(F(p) \wedge B(p)) \rightarrow \exists jL(j)$
 b) $\forall pB(p) \rightarrow \exists jQ(j)$
 c) $\exists j(Q(j) \wedge L(j)) \rightarrow \exists pF(p)$
 d) $(\forall pB(p) \wedge \forall jQ(j)) \rightarrow \exists jL(j)$
40. Express each of these system specifications using predicates, quantifiers, and logical connectives.
 a) When there is less than 30 megabytes free on the hard disk, a warning message is sent to all users.
 b) No directories in the file system can be opened and no files can be closed when system errors have been detected.
 c) The file system cannot be backed up if there is a user currently logged on.
 d) Video on demand can be delivered when there are at least 8 megabytes of memory available and the connection speed is at least 56 kilobits per second.

41. Express each of these system specifications using predicates, quantifiers, and logical connectives.
 a) At least one mail message, among the nonempty set of messages, can be saved if there is a disk with more than 10 kilobytes of free space.
 b) Whenever there is an active alert, all queued messages are transmitted.
 c) The diagnostic monitor tracks the status of all systems except the main console.
 d) Each participant on the conference call whom the host of the call did not put on a special list was billed.
42. Express each of these system specifications using predicates, quantifiers, and logical connectives.
 a) Every user has access to an electronic mailbox. b) The system mailbox can be accessed by everyone in the group if the file system is locked.
 c) The firewall is in a diagnostic state only if the proxy server is in a diagnostic state.
 d) At least one router is functioning normally if the throughput is between 100 kbps and 500 kbps and the proxy server is not in diagnostic mode.

56 1 / The Foundations: Logic and Proofs

43. Determine whether $\forall x(P(x) \rightarrow Q(x))$ and $\forall xP(x) \rightarrow \forall xQ(x)$ are logically equivalent. Justify your answer.
 44. Determine whether $\forall x(P(x) \leftrightarrow Q(x))$ and $\forall xP(x) \leftrightarrow \forall xQ(x)$ are logically equivalent. Justify your answer.
 45. Show that $\exists x(P(x) \vee Q(x))$ and $\exists xP(x) \vee \exists xQ(x)$ are logically equivalent.

Exercises 46–49 establish rules for **null quantification** that we can use when a quantified variable does not appear in part of a statement.

46. Establish these logical equivalences, where x does not occur as a free variable in A . Assume that the domain is nonempty.
 a) $(\forall xP(x)) \vee A \equiv \forall x(P(x) \vee A)$
 b) $(\exists xP(x)) \vee A \equiv \exists x(P(x) \vee A)$
47. Establish these logical equivalences, where x does not occur as a free variable in A . Assume that the domain is nonempty.
 a) $(\forall xP(x)) \wedge A \equiv \forall x(P(x) \wedge A)$
 b) $(\exists xP(x)) \wedge A \equiv \exists x(P(x) \wedge A)$
48. Establish these logical equivalences, where x does not occur as a free variable in A . Assume that the domain is nonempty.
 a) $\forall x(A \rightarrow P(x)) \equiv A \rightarrow \forall xP(x)$
 b) $\exists x(A \rightarrow P(x)) \equiv A \rightarrow \exists xP(x)$
49. Establish these logical equivalences, where x does not occur as a free variable in A . Assume that the domain is nonempty.
 a) $\forall x(P(x) \rightarrow A) \equiv \exists xP(x) \rightarrow A$
 b) $\exists x(P(x) \rightarrow A) \equiv \forall xP(x) \rightarrow A$
50. Show that $\forall xP(x) \vee \forall xQ(x)$ and $\forall x(P(x) \vee Q(x))$ are not logically equivalent.
 51. Show that $\exists xP(x) \wedge \exists xQ(x)$ and $\exists x(P(x) \wedge Q(x))$ are not logically equivalent.
 52. As mentioned in the text, the notation $\exists!xP(x)$ denotes “There exists a unique x such that $P(x)$ is

true.” If the domain consists of all integers, what are the truth values of these statements?

- a) $\exists !x(x > 1)$ b) $\exists !x(x^2 = 1)$
c) $\exists !x(x + 3 = 2x)$ d) $\exists !x(x = x + 1)$ 53.

What are the truth values of these statements? a)

$\exists !xP(x) \rightarrow \exists xP(x)$

- b) $\forall xP(x) \rightarrow \exists !xP(x)$
c) $\exists !x\neg P(x) \rightarrow \neg \forall xP(x)$

54. Write out $\exists !xP(x)$, where the domain consists of the integers 1, 2, and 3, in terms of negations, conjunctions, and disjunctions.

55. Given the Prolog facts in Example 28, what would Prolog return given these queries?

- a) `?instructor(chan, math273)`
b) `?instructor(patel, cs301)`
c) `?enrolled(X, cs301)`
d) `?enrolled(kiko, Y)`
e) `?teaches(grossman, Y)`

56. Given the Prolog facts in Example 28, what would Prolog return when given these queries?

- a) `?enrolled(kevin, ee222)`
b) `?enrolled(kiko, math273)`
c) `?instructor(grossman, X)`
d) `?instructor(X, cs301)`
e) `?teaches(X, kevin)`

57. Suppose that Prolog facts are used to define the predicates *mother*(*M*, *Y*) and *father*(*F*, *X*), which represent that *M* is the mother of *Y* and *F* is the father of *X*, respectively. Give a Prolog rule to define the predicate *sibling*(*X*, *Y*), which represents that *X* and *Y* are siblings (that is, have the same mother and the same father).

58. Suppose that Prolog facts are used to define the predicates *mother*(*M*, *Y*) and *father*(*F*, *X*), which represent that *M* is the mother of *Y* and *F* is the father of *X*, respectively. Give a Prolog rule to define the predicate *grandfather*(*X*, *Y*), which represents that *X* is the grand father of *Y*. [Hint: You can write a disjunction in Prolog either by using a semicolon to separate predicates or by putting these predicates on separate lines.]

Exercises 59–62 are based on questions found in the book *Symbolic Logic* by Lewis Carroll.

59. Let *P*(*x*), *Q*(*x*), and *R*(*x*) be the statements “*x* is a professor,” “*x* is ignorant,” and “*x* is vain,” respectively. Express each of these statements using quantifiers; logical connectives; and *P*(*x*), *Q*(*x*), and *R*(*x*), where the domain consists of all people.

- a) No professors are ignorant.
b) All ignorant people are vain.
c) No professors are vain.
d) Does (c) follow from (a) and (b)?

60. Let *P*(*x*), *Q*(*x*), and *R*(*x*) be the statements “*x* is a clear explanation,” “*x* is satisfactory,” and “*x* is an excuse,” respectively. Suppose that the domain for *x* consists of all English text. Express each of these statements using quantifiers, logical connectives, and *P*(*x*), *Q*(*x*), and *R*(*x*).

- a) All clear explanations are satisfactory.
b) Some excuses are unsatisfactory.
c) Some excuses are not clear explanations.

* d) Does (c) follow from (a) and (b)?

61. Let *P*(*x*), *Q*(*x*), *R*(*x*), and *S*(*x*) be the statements “*x* is a baby,” “*x* is logical,” “*x* is able to manage a crocodile,” and “*x* is despised,” respectively. Suppose that the domain consists of all people. Express each of these statements using quantifiers; logical connectives; and *P*(*x*), *Q*(*x*), *R*(*x*), and *S*(*x*).

- a) Babies are illogical.
b) Nobody is despised who can manage a crocodile. c) Illogical persons are despised.
d) Babies cannot manage crocodiles.

* e) Does (d) follow from (a), (b), and (c)? If not, is there a correct conclusion?

62. Let *P*(*x*), *Q*(*x*), *R*(*x*), and *S*(*x*) be the statements “*x* is a duck,” “*x* is one of my poultry,” “*x* is an officer,” and “*x* is willing to waltz,” respectively. Express each of these statements using quantifiers; logical connectives; and *P*(*x*), *Q*(*x*), *R*(*x*), and *S*(*x*).

a) No ducks are willing to waltz.

Nested Quantifiers

Introduction

1.5 Nested Quantifiers 57

- b) No officers ever decline to waltz.
c) All my poultry are ducks.
d) My poultry are not officers.

* e) Does (d) follow from (a), (b), and (c)? If not, is there a correct conclusion?

In Section 1.4 we defined the existential and universal quantifiers and showed how they can be used to represent mathematical statements. We also explained how they can be used to

translate English sentences into logical expressions. However, in Section 1.4 we avoided **nested quantifiers**, where one quantifier is within the scope of another, such as

$$\forall x \exists y (x + y = 0).$$

Note that everything within the scope of a quantifier can be thought of as a propositional function. For example,

$$\forall x \exists y (x + y = 0)$$

is the same thing as $\forall x Q(x)$, where $Q(x)$ is $\exists y P(x, y)$, where $P(x, y)$ is $x + y = 0$. Nested quantifiers commonly occur in mathematics and computer science. Although nested quantifiers can sometimes be difficult to understand, the rules we have already studied in Section 1.4 can help us use them. In this section we will gain experience working with nested quantifiers. We will see how to use nested quantifiers to express mathematical statements such as “The sum of two positive integers is always positive.” We will show how nested quantifiers can be used to translate English sentences such as “Everyone has exactly one best friend” into logical statements. Moreover, we will gain experience working with the negations of statements involving nested quantifiers.

Understanding Statements Involving Nested Quantifiers

To understand statements involving nested quantifiers, we need to unravel what the quantifiers and predicates that appear mean. This is illustrated in Examples 1 and 2.

EXAMPLE 1 Assume that the domain for the variables x and y consists of all real numbers. The statement

$$\forall x \forall y (x + y = y + x)$$

says that $x + y = y + x$ for all real numbers x and y . This is the commutative law for addition of real numbers. Likewise, the statement

$$\forall x \exists y (x + y = 0)$$

says that for every real number x there is a real number y such that $x + y = 0$. This states that every real number has an additive inverse. Similarly, the statement

$$\forall x \forall y \forall z (x + (y + z) = (x + y) + z)$$

is the associative law for addition of real numbers. .

EXAMPLE 2 Translate into English the statement

$$\forall x \forall y ((x > 0) \wedge (y < 0) \rightarrow (xy < 0)),$$

where the domain for both variables consists of all real numbers.

Solution: This statement says that for every real number x and for every real number y , if $x > 0$ and $y < 0$, then $xy < 0$. That is, this statement says that for real numbers x and y , if x is positive and y is negative, then xy is negative. This can be stated more succinctly as “The product of a positive real number and a negative real number is always a negative real number.” .

THINKING OF QUANTIFICATION AS LOOPS In working with quantifications of more than one variable, it is sometimes helpful to think in terms of nested loops. (Of course, if there are infinitely many elements in the domain of some variable, we cannot actually loop through all values. Nevertheless, this way of thinking is helpful in understanding nested quantifiers.) For example, to see whether $\forall x \forall y P(x, y)$ is true, we loop through the values for x , and for

each x we loop through the values for y . If we find that $P(x, y)$ is true for all values for x and y , we have determined that $\forall x \forall y P(x, y)$ is true. If we ever hit a value x for which we hit a value y for which $P(x, y)$ is false, we have shown that $\forall x \forall y P(x, y)$ is false.

Similarly, to determine whether $\forall x \exists y P(x, y)$ is true, we loop through the values for x . For each x we loop through the values for y until we find a y for which $P(x, y)$ is true. If for every x we hit such a y , then $\forall x \exists y P(x, y)$ is true; if for some x we never hit such a y , then $\forall x \exists y P(x, y)$ is false.

To see whether $\exists x \forall y P(x, y)$ is true, we loop through the values for x until we find an x for which $P(x, y)$ is always true when we loop through all values for y . Once we find such an x , we know that $\exists x \forall y P(x, y)$ is true. If we never hit such an x , then we know that $\exists x \forall y P(x, y)$ is false.

Finally, to see whether $\exists x \exists y P(x, y)$ is true, we loop through the values for x , where for each x we loop through the values for y until we hit an x for which we hit a y for which $P(x, y)$ is true. The statement $\exists x \exists y P(x, y)$ is false only if we never hit an x for which we hit a y such that $P(x, y)$ is true.

The Order of Quantifiers

Many mathematical statements involve multiple quantifications of propositional functions involving more than one variable. It is important to note that the order of the quantifiers is important, unless all the quantifiers are universal quantifiers or all are existential quantifiers.

These remarks are illustrated by Examples 3–5.

EXAMPLE 3 Let $P(x, y)$ be the statement “ $x + y = y + x$.” What are the truth values of the quantifications $\forall x \forall y P(x, y)$ and $\forall y \forall x P(x, y)$ where the domain for all variables consists of all real numbers?

Solution: The quantification

$$\forall x \forall y P(x, y)$$

denotes the proposition

“For all real numbers x , for all real numbers y , $x + y = y + x$.”

Because $P(x, y)$ is true for all real numbers x and y (it is the commutative law for addition, which is an axiom for the real numbers—see Appendix 1), the proposition $\forall x \forall y P(x, y)$ is true. Note that the statement $\forall y \forall x P(x, y)$ says “For all real numbers y , for all real numbers x , $x + y = y + x$.” This has the same meaning as the statement “For all real numbers x , for all real numbers y , $x + y = y + x$.” That is, $\forall x \forall y P(x, y)$ and $\forall y \forall x P(x, y)$ have the same meaning,

1.5 Nested Quantifiers 59

and both are true. This illustrates the principle that the order of nested universal quantifiers in a statement without other quantifiers can be changed without changing the meaning of the quantified statement.

EXAMPLE 4 Let $Q(x, y)$ denote “ $x + y = 0$.” What are the truth values of the quantifications $\exists y \forall x Q(x, y)$ and $\forall x \exists y Q(x, y)$, where the domain for all variables consists of all real numbers?

Solution: The quantification

$$\exists y \forall x Q(x, y)$$

denotes the proposition

“There is a real number y such that for every real number x , $Q(x, y)$.”

No matter what value of y is chosen, there is only one value of x for which $x + y = 0$. Because there is no real number y such that $x + y = 0$ for all real numbers x , the statement $\exists y \forall x Q(x, y)$ is false.

The quantification

$$\forall x \exists y Q(x, y)$$

denotes the proposition

“For every real number x there is a real number y such that $Q(x, y)$.”

Given a real number x , there is a real number y such that $x + y = 0$; namely, $y = -x$. Hence, the statement $\forall x \exists y Q(x, y)$ is true. [Be careful with the order](#)

[of existential and universal quantifiers!](#)

Example 4 illustrates that the order in which quantifiers appear makes a difference. The statements $\exists y \forall x P(x, y)$ and $\forall x \exists y P(x, y)$ are not logically equivalent. The statement $\exists y \forall x P(x, y)$ is true if and only if there is a y that makes $P(x, y)$ true for every x . So, for this statement to be true, there must be a particular value of y for which $P(x, y)$ is true regardless of the choice of x . On the other hand, $\forall x \exists y P(x, y)$ is true if and only if for every value of x there is a value of y for which $P(x, y)$ is true. So, for this statement to be true, no matter which x you choose, there must be a value of y (possibly depending on the x

you choose) for which $P(x, y)$ is true. In other words, in the second case, y can depend on x , whereas in the first case, y is a constant independent of x .

From these observations, it follows that if $\exists y \forall x P(x, y)$ is true, then $\forall x \exists y P(x, y)$ must also be true. However, if $\forall x \exists y P(x, y)$ is true, it is not necessary for $\exists y \forall x P(x, y)$ to be true. (See Supplementary Exercises 30 and 31.)

Table 1 summarizes the meanings of the different possible quantifications involving two variables.

Quantifications of more than two variables are also common, as Example 5 illustrates.

EXAMPLE 5 Let $Q(x, y, z)$ be the statement “ $x + y = z$.” What are the truth values of the statements $\forall x \forall y \exists z Q(x, y, z)$ and $\exists z \forall x \forall y Q(x, y, z)$, where the domain of all variables consists of all real numbers?

Solution: Suppose that x and y are assigned values. Then, there exists a real number z such that $x + y = z$. Consequently, the quantification

$$\forall x \forall y \exists z Q(x, y, z),$$

which is the statement

“For all real numbers x and for all real numbers y there is a real number z such that $x + y = z$,”