
Boolean Reasoning

Boolean Reasoning
The Logic of Boolean Equations

by

Frank Markham Brown
Air Force Institute of Technology



Springer Science+Business Media, LLC

Consulting Editor: Jonathan Allen

Library of Congress Cataloging-in-Publication Data

Brown, Frank Markham, 1930-

Boolean reasoning / by Frank Markham Brown.

p. cm.

Includes bibliographical references (p. 247-264).

Includes index.

ISBN 978-1-4757-2080-8

ISBN 978-1-4757-2078-5 (eBook)

DOI 10.1007/978-1-4757-2078-5

1. Algebra, Boolean. I. Title.

QA10.3.B76 1990

511.3 '24—dc20

90-4714

CIP

Copyright © 1990 by Springer Science+Business Media New York

Originally published by Kluwer Academic Publishers in 1990

Softcover reprint of the hardcover 1st edition 1990

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher, Springer Science+Business Media, LLC

Contents

Preface	xi
Two Logical Languages	xi
Boolean Reasoning	xii
Boolean Algebra and Switching Theory	xiii
An Approach to Boolean Problem-Solving	xiv
Boolean Reasoning vs. Predicate Logic	xv
Outline	xvi
Acknowledgments	xviii
1 Fundamental Concepts	1
1.1 Formulas	1
1.2 Propositions and Predicates	2
1.3 Sets	5
1.4 Operations on Sets	9
1.5 Partitions	11
1.6 Relations	11
1.7 Functions	16
1.8 Operations and Algebraic Systems	18
2 Boolean Algebras	23
2.1 Postulates for a Boolean Algebra	23
2.2 Examples of Boolean Algebras	24
2.2.1 The Algebra of Classes (Subsets of a Set)	24
2.2.2 The Algebra of Propositional Functions	25
2.2.3 Arithmetic Boolean Algebras	26
2.2.4 The Two-Element Boolean Algebra	26
2.2.5 Summary of Examples	26
2.3 The Stone Representation Theorem	27

2.4	The Inclusion-Relation	28
2.4.1	Intervals	30
2.5	Some Useful Properties	30
2.6	n-Variable Boolean Formulas	33
2.7	n-Variable Boolean Functions	34
2.8	Boole's Expansion Theorem	36
2.9	The Minterm Canonical Form	39
2.9.1	Truth-tables	41
2.9.2	Maps	42
2.10	The Löwenheim-Müller Verification Theorem	44
2.11	Switching Functions	45
2.12	Incompletely-Specified Boolean Functions	45
2.13	Boolean Algebras of Boolean Functions	47
2.13.1	Free Boolean Algebras	48
2.14	Orthonormal Expansions	48
2.14.1	Löwenheim's Expansions	50
2.15	Boolean Quotient	53
2.16	The Boolean Derivative	56
2.17	Recursive Definition of Boolean Functions	58
2.18	What Good are "Big" Boolean Algebras?	60
3	The Blake Canonical Form	71
3.1	Definitions and Terminology	72
3.2	Syllogistic & Blake Canonical Formulas	73
3.3	Generation of $BCF(f)$	75
3.4	Exhaustion of Implicants	76
3.5	Iterated Consensus	77
3.5.1	Quine's method	78
3.5.2	Successive extraction	80
3.6	Multiplication	80
3.6.1	Recursive multiplication	81
3.6.2	Combining multiplication and iterated consensus	83
3.6.3	Unwanted syllogistic formulas	85

4 Boolean Analysis 87

- 4.1 Review of Elementary Properties 87
- 4.2 Boolean Systems 88
 - 4.2.1 Antecedent, Consequent, and Equivalent Systems . . . 89
 - 4.2.2 Solutions 89
- 4.3 Reduction 89
- 4.4 The Extended Verification Theorem 91
- 4.5 Poretsky’s Law of Forms 92
- 4.6 Boolean Constraints 93
- 4.7 Elimination 95
- 4.8 Eliminants 100
- 4.9 Rudundant Variables 107
- 4.10 Substitution 113
- 4.11 The Tautology Problem 115
 - 4.11.1 Testing for Tautology 115
 - 4.11.2 The Sum-to-One Theorem 116
 - 4.11.3 Nearly-Minimal SOP Formulas 117

5 Syllogistic Reasoning 123

- 5.1 The Principle of Assertion 124
- 5.2 Deduction by Consensus 126
- 5.3 Syllogistic Formulas 127
- 5.4 Clausal Form 129
- 5.5 Producing and Verifying Consequents 132
 - 5.5.1 Producing Consequents 132
 - 5.5.2 Verifying Consequents 133
 - 5.5.3 Comparison of Clauses 134
- 5.6 Class-Logic 134
- 5.7 Selective Deduction 136
- 5.8 Functional Relations 138
- 5.9 Dependent Sets of Functions 140
- 5.10 Sum-to-One Subsets 143
- 5.11 Irredundant Formulas 145

6	Solution of Boolean Equations	153
6.1	Particular Solutions and Consistency	154
6.2	General Solutions	156
6.3	Subsumptive General Solutions	158
6.3.1	Successive Elimination	159
6.3.2	Deriving Eliminants from Maps	161
6.3.3	Recurrent Covers and Subsumptive Solutions	162
6.3.4	Simplified Subsumptive Solutions	166
6.3.5	Simplification via Marquand Diagrams	167
6.4	Parametric General Solutions	167
6.4.1	Successive Elimination	169
6.4.2	Parametric Solutions based on Recurrent Covers	172
6.4.3	Löwenheim's Formula	175
7	Functional Deduction	181
7.1	Functionally Deducible Arguments	182
7.2	Eliminable and Determining Subsets	187
7.2.1	u -Eliminable Subsets	187
7.2.2	u -Determining Subsets	189
7.2.3	Calculation of Minimal u -Determining Subsets	190
8	Boolean Identification	193
8.1	Parametric and Diagnostic Models	195
8.1.1	Parametric Models	195
8.1.2	The Diagnostic Axiom	197
8.1.3	Diagnostic Equations and Functions	197
8.1.4	Augmentation	199
8.2	Adaptive Identification	201
8.2.1	Initial and Terminal Specifications	201
8.2.2	Updating the Model	204
8.2.3	Effective Inputs	205
8.2.4	Test-Procedure	208

- 9 Recursive Realizations of Combinational Circuits 211**
 - 9.1 The Design-Process 212
 - 9.2 Specifications 213
 - 9.2.1 Specification-Formats 214
 - 9.2.2 Consistent Specifications 218
 - 9.3 Tabular Specifications 219
 - 9.4 Strongly Combinational Solutions 223
 - 9.5 Least-Cost Recursive Solutions 224
 - 9.6 Constructing Recursive Solutions 229
 - 9.6.1 The Procedure 232
 - 9.6.2 An Implementation using BORIS 234

- A Syllogistic Formulas 239**
 - A.1 Absorptive Formulas 240
 - A.2 Syllogistic Formulas 240
 - A.3 Prime Implicants 244
 - A.4 The Blake Canonical Form 245

- Bibliography 247**

- Index 265**

Preface

This book is about the logic of Boolean equations. Such equations were central in the “algebra of logic” created in 1847 by Boole [12, 13] and developed by others, notably Schröder [178], in the remainder of the nineteenth century. Boolean equations are also the language by which digital circuits are described today.

Logicians in the twentieth century have abandoned Boole’s equation-based logic in favor of the more powerful predicate calculus. As a result, digital engineers—and others who use Boole’s language routinely—remain largely unaware of its utility as a medium for reasoning. The aim of this book, accordingly, is to present a systematic outline of the logic of Boolean equations, in the hope that Boole’s methods may prove useful in solving present-day problems.

Two Logical Languages

Logic seeks to reduce reasoning to calculation. Two main languages have been developed to achieve that object: Boole’s “algebra of logic” and the predicate calculus. Boole’s approach was to represent classes (*e.g.*, happy creatures, things productive of pleasure) by symbols and to represent logical statements as equations to be solved. His formulation proved inadequate, however, to represent ordinary discourse. A number of nineteenth-century logicians, including Jevons [94], Poretsky [159], Schröder [178], Venn [210], and Whitehead [212, 213], sought an improved formulation based on extensions or modifications of Boole’s algebra. These efforts met with only limited success. A different approach was taken in 1879 by Frege [60], whose system was the ancestor of the predicate calculus. The latter language has superseded Boolean algebra as a medium for general symbolic reasoning.

The elementary units of discourse in the predicate calculus are the *predicates*, or *atomic formulas*. These are statements such as “X likes Mary,” and “ $X > Y + 2$,” which, for any allowed values of their variables, are either true or false. The variables in a predicate may be *quantified*, by the symbols \forall (“for all”) or \exists (“there exists”), to form statements such as $\forall X(X \text{ likes Mary})$ or $\exists X(X \text{ likes Mary})$; these statements mean, respectively, “everyone likes Mary” and “someone likes Mary.” Predicates may be assembled into more complex structures, called *well-formed formulas*, by means of logical connectives such as conjunction (AND), disjunction (OR), and negation (NOT).

Boolean Reasoning

Boolean reasoning builds on the Boole-Schröder algebra of logic, which is based on Boolean equations, rather than on the predicate calculus. Although Boolean equations are predicates—statements that are either true or false for any values of their arguments—almost none of the apparatus of predicate logic is employed in Boolean reasoning. Neither the disjunction of two Boolean equations nor the negation of a Boolean equation is a Boolean equation; thus neither of these operations is generally allowed in Boolean reasoning (see Rudeanu [172, Chapt. 10] for results concerning disjunction and negation of Boolean equations). As shown by Boole, however, the conjunction of two or more Boolean equations is a Boolean equation. The conjunction of a system of equations is therefore expressed by its equivalent single equation, rather than by a symbolic conjunction of equations. Thus the only well-formed formulas of interest in Boolean reasoning are Boolean equations.

Boole and other nineteenth-century logicians based symbolic reasoning on an equation of the *0-normal* form, *i.e.*,

$$f(x_1, \dots, x_n) = 0, \quad (1)$$

derived from, and equivalent to, a given system of Boolean equations (the equivalent *1-normal* form, *i.e.*, $f'(x_1, \dots, x_n) = 1$, may also be used). A dissertation published in 1937 by A. Blake [10] showed that the consequents of (1) are readily derived from the *prime implicants* of f . The concept of a prime implicant was re-discovered (and named) in 1952 by W.V.O. Quine, who investigated the problem of minimizing the complexity of Boolean formulas. Quine established the theoretical foundations of minimization-theory

in a series of papers [161, 162, 163, 164] in the 1950s. The theory of prime implicants has thus arisen independently to serve two quite different ends, *viz.*, Boolean reasoning (Blake) and formula-minimization (Quine).

The approach to Boolean reasoning outlined in this book owes much to Blake's work. Blake's formulation (outlined in Appendix A) anticipates, within the domain of Boolean algebra, the widely-applied resolution principle in predicate logic, given in 1965 by Robinson [168]. Blake's "syllogistic result," for example, corresponds to Robinson's "resolvent."

Boolean Algebra and Switching Theory

Although Boole's algebra did not succeed in expressing, as he had intended, "those operations of the mind by which reasoning is performed" [13, p. 1], it remains in daily use to deal with the simpler mentality of switching circuits. The possibility of applying Boolean algebra to the design of switching systems was first suggested in 1910 by the physicist P. Ehrenfest [54], who proposed in a review of a text by Couturat [41] that Boolean algebra be used in the design of automatic telephone exchanges. Ehrenfest did not, however, supply details as to how it might be done. Papers providing such details appeared independently between 1936 and 1938 in Japan, the United States, and the Soviet Union (it seems that only the results published in the Soviet Union, by Shestakov [185], were based on Ehrenfest's suggestion). The most influential of these papers was Shannon's "Symbolic Analysis of Relay and Switching Circuits" [183], based on his M.S. thesis at the Massachusetts Institute of Technology. Shannon formulated a "calculus of switching circuits," which he showed to be analogous to the calculus of propositions. In a paper published in Japan in 1937, Nakasima [146] identified the same switching calculus with the algebra of sets. Nakasima's paper, the earliest to apply Boolean algebra to switching theory, discussed methods of solving Boolean equations, with the aim of finding an unknown component switching-path when the composite path is known. Nakasima's work seems to have been little noticed in the United States; in the 1950s, however, a number of papers appeared in the U.S. which applied Boolean equation-solving to the design of switching systems [2, 4, 113, 155].

Motivated by problems arising in the design of switching circuits, A. Svoboda [191] proposed construction of a "Boolean Analyzer," a hardware-adjunct to a general-purpose computer, specialized to solve Boolean equations. The applications of such a unit, and of APL programs for solving

Boolean equations, are described in Svoboda and White [193].

Klir and Marin [103] have stated that “The most powerful tool of the modern methodology of switching circuits seems to be the Boolean equations. Their importance for switching theory reminds one of the application of differential equations in electric circuit theory.” There remains a curious difference, however, between the way differential equations and Boolean equations are typically applied: Boolean equations are rarely solved. They are manipulated in form but are seldom the basis for systematic reasoning. Contemporary research on Boolean methods in switching tends instead to emphasize formula-minimization. Many writers on applications of Boolean methods believe in fact that the only useful thing to do with Boolean formulas is to simplify them. A widely-used text [84, p. 60] announces that “Almost every problem in Boolean algebra will be found to be some variation of the following statement: ‘Given one of the 2^{2^n} functions of n variables, determine from the large number of equivalent expressions of this function one which satisfies some criteria for simplicity.’ ” Boole would doubtless deem that to be less than full employment for the algebra he designed as an instrument for reasoning.

Although the processes of Boolean reasoning should for practical reasons keep their internal representations relatively simple, minimization is a topic essentially distinct from reasoning. Minimization is therefore not emphasized in this book, notwithstanding its importance both theoretically and in practice. See Brayton, *et al.*, [18] for an excellent contemporary treatment of minimization and its applications in the design of VLSI (Very Large-Scale Integration) circuits.

An Approach to Boolean Problem-Solving

The central idea in Boolean reasoning, first given by Boole, is to reduce a given system of logical equations to a single equivalent equation of standardized form (*e.g.*, $f = 0$), and then to carry out the desired reasoning on that equation. This preliminary abstraction enables the processes of reasoning to be independent of the form of the original equations.

The primary tactic employed by Boole and later nineteenth-century logicians was to solve $f(x, y, \dots) = 0$ for certain of its arguments in terms of others. A solution of an equation is a particular kind of antecedent, however, and not necessarily a consequent, of the equation. An important advance was made in 1937 by Blake [10], who showed that the consequents of $f = 0$

are represented economically in the disjunction of the prime implicants of f . We call this disjunction the *Blake canonical form* for f and denote it by $BCF(f)$.

The task of solving a problem based on a collection of Boolean equations may thus be carried out in three major steps:

1. **Reduction.** Condense the equations into a single Boolean equation of the form $f = 0$.
2. **Development.** Construct the Blake canonical form for f , *i.e.*, generate the prime implicants of f .
3. **Reasoning.** Apply a sequence of reasoning-operations, beginning with the Blake form, to solve the problem.

Steps 1 and 2 are independent of the problem to be solved and are readily automated. The sequence of operations to be applied in Step 3, on the other hand, is dependent upon the problem. To employ Boolean reasoning, therefore, the principal task is to select an appropriate sequence of operations to apply to the formula $BCF(f)$. The operational (*i.e.*, functional) basis of Boolean reasoning differentiates it from the predicate calculus, whose basis is relational. Other differences between the two languages are discussed below.

Boolean Reasoning *vs.* Predicate Logic

The need to incorporate systematic reasoning into the design of switching systems has attracted the attention of engineers to the theorem-proving methods of the predicate calculus. Design based on these methods is summarized by Kabat and Wojcik [95] as follows: "The basic philosophy of the design approach using theorem proving is to represent the elements of the design process as a set of axioms in a formal system (a theory), state the problem of realizability of the target function as a theorem, and prove it in the context of the theory. Once the theorem is proved, an automatic procedure for the recovery of the logic circuit is to be executed to complete the design."

Boolean reasoning differs from the theorem-proving methodology of predicate logic in a number of important ways. Predicates (propositional functions) and propositions are two-valued. Boolean functions, on the other hand, take on values over an underlying set, the *carrier* of the associated

Boolean algebra; the number of elements in the carrier may be 2 or any higher power of 2. The following properties—valid in propositional logic—do not hold in other than two-valued Boolean algebras:

$$\begin{aligned} F + G = 1 &\implies F = 1 \quad \text{or} \quad G = 1 \\ F \neq 1 &\implies F = 0. \end{aligned}$$

Applying propositional calculation-rules to Boolean problems can therefore lead to incorrect results. The denial of a biconditional in propositional logic, for example, can be expressed as a biconditional; thus $\neg(a \longleftrightarrow b)$ is equivalent to $a \longleftrightarrow \neg b$. The denial of a Boolean equation, however, cannot in general be expressed as a Boolean equation; denying the Boolean equation $a = b$ is not the same as asserting the equation $a = b'$.

The principal problem-solving technique in predicate logic is theorem-proving via refutation, *i.e.*, *reductio ad absurdum*. This technique entails the denial of the theorem to be proved. As noted above, however, the denial of a Boolean equation is not a Boolean equation, and thus refutation-based reasoning is not possible in Boolean algebras having other than two values.

Problem-solving in predicate logic entails assigning values to variables over some domain. Any set, *e.g.*, $\{5, \text{John}, \clubsuit, \text{cat}, \star\}$, may be the domain for a problem in predicate logic. The domain in a Boolean problem, however, is an ordered structure—the carrier of a Boolean algebra. Consequently, the information produced by Boolean reasoning is typically expressed by intervals (cf. Section 2.4.1).

Outline

Chapters 1 through 4 of this book outline the mathematical basis for Boolean reasoning. Chapter 1, included to make the book self-contained, is a brief survey of fundamental concepts such as propositions, predicates, sets, relations, functions and algebraic systems. Chapter 2 treats the classical Boole-Schröder algebra of logic via Huntington's postulates. Several examples of Boolean algebras are discussed, and important theorems are presented; among the latter are the Stone representation theorem, Boole's expansion theorem, and the Löwenheim-Müller verification theorem. Boolean formulas and Boolean functions are defined and the distinction between these two concepts is emphasized. Orthonormal expansions are defined and their utility is examined. The utility of "big" Boolean algebras (those comprising more than two elements) is discussed. Chapter 3 outlines Blake's theory of

canonical formulas [10], and the employment of such formulas in deriving and verifying consequents of Boolean equations. Several methods are presented for constructing the Blake canonical form. The Blake form is employed frequently in the remainder of the book; a number of theorems concerning this form, based on Blake's dissertation [10], are given in Appendix A. Chapter 4 introduces the basic operations from which reasoning procedures may be composed. Among such operations are reduction, elimination, expansion, division, and substitution.

Chapters 5 through 7 treat two categories of Boolean reasoning: syllogistic (Chapter 5) and functional (Chapters 6 and 7). Syllogistic reasoning, a direct approach to the solution of problems in propositional logic, is based on constructing a simplified representation of the consequents of the Boolean equation $f = 0$. Functional reasoning, on the other hand, produces functional equations, *i.e.*, statements of the form $x = g(y, z, \dots)$, related to the equation $f(x, y, z, \dots) = 0$. Functional antecedents, solutions of $f = 0$, were investigated by Boole and have been the object of much study since; see Rudeanu [172] for an authoritative survey and a complete bibliography. Functional consequents, on the other hand, seem to have received little attention; we discuss the theory of such consequents as well as a number of their applications.

The last two chapters present applications of Boolean reasoning in digital technology. Chapter 8 discusses the identification of a Boolean "black box" by means of an input-output experiment. Chapter 9 concerns multiple-output combinational switching circuits. Emphasis is placed on the problem of specification; the design-problem is formulated as one of solving the specification. A particular class of solutions, which we call *recursive*, corresponds to loop-free circuits which may employ output-signals to help in generating other output-signals.

Proofs are supplied for new results; a proof is given for an established result, however, only if it is particularly instructive. The Boolean calculations entailed in the examples—and those underlying the new results—were carried out using a set of software-tools which the author calls BORIS (Boolean Reasoning In Scheme); these tools are programmed in PC Scheme, a microcomputer-based dialect of Lisp available from Texas Instruments, Inc. BORIS has been invaluable in the exploration and testing of conjectures, building confidence in good conjectures and rudely puncturing bad ones.

Acknowledgments

The author's interest in Boolean methods owes much to the example and assistance of Professor Sergiu Rudeanu of the Faculty of Mathematics, University of Bucharest, Romania. I am indebted to Professor Rudeanu for his careful reading of a preliminary version of this book; full many a logical rock and mathematical shoal was avoided thanks to his comments. I am indebted also to Captain James J. Kainec, U.S. Army, for reading several generations of this book and supplying a large number of helpful comments. Several chapters were read by Dr. Albert W. Small, who made useful suggestions and corrections.

I wish finally to acknowledge a debt beyond measure—to my dear wife, Roberta. Her steadfast support and encouragement made this book possible.

Boolean Reasoning