Logic Functions and Equations

Christian Posthoff • Bernd Steinbach

Logic Functions and Equations

Binary Models for Computer Science

Second Edition

Springer

Christian Posthoff
Computing and Information Technology
University of the West Indies (retired)
Chemnitz, Sachsen
Germany

Bernd Steinbach
Computer Science
TU Bergakademie Freiberg (retired)
Chemnitz, Sachsen
Germany

ISBN 978-3-030-02419-2 ISBN 978-3-030-02420-8 (eBook)
https://doi.org/10.1007/978-3-030-02420-8

Library of Congress Control Number: 2018959447

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

Foreword

Since the introduction of Boolean Algebras, two different areas have been developed for their applications. One is mathematical logic, and the other is switching theory. Mathematical logic is used for the proof of the theory, while switching theory is used for the analysis and synthesis of logic circuits.

Logic functions, the central topic of this book, are commonly used in these two areas. Such functions are also used in logic equations which very universally facilitate the specification of many problems to solve. This book extends these means of expression especially by derivative operations of the Boolean Differential Calculus as well as Boolean differential equations. Derivative operations are even determined for lattices of logic functions and successfully applied for the synthesis of combinational circuits by decompositions.

A satisfiability (SAT) equation is a special type of a logic equation. Recently, with the tremendous advancement of SAT solvers, some logic design systems use SAT solvers to optimize circuits. Furthermore, the SAT solvers have become a standard method to solve combinatorial problems, e.g., graph coloring, graph enumeration, assignment problems, and puzzles. Although dedicated algorithms can solve these problems more efficiently, SAT solvers are often used to solve them. One reason is to develop a dedicated algorithm for each problem is often too complex and too expensive. Another reason is that modern computers and software are efficient enough to solve these problems in a reasonable computation time. Thus, SAT solvers are now routinely used in such applications.

Combinatorial problems that appear in engineering are often NP-hard, and they are often solved by so-called meta-heuristics such as genetic algorithm, tabu search, and simulated annealing. The merit of the SAT-based method is the ability to find an optimal solution.

This book offers many interesting applications: bent functions, functions that require the largest number of product terms in ESOP, graph coloring problems, covering problems, Hamiltonian paths, Eulerian paths, knight tours, queens problem, bishops problem, Ramsey theory, Latin squares, Sudoku, and Pythagorean triples, in addition to optimization problems of logic circuits. It also shows limitations of

SAT solvers: four-colored grid problems which are very hard to solve. Most results come from long research efforts of the authors.

I assume that readers can find interesting and useful applications in this unique book.

Kanagawa, Japan
August 2018

Tsutomu Sasao

Preface

Two more or less independent areas existed when we started our cooperative research in the area of binary systems that is covered by this textbook. One part is very much related to Mathematics or can even be considered as a fundamental tool for the construction of *axiomatic Mathematics* altogether. This part goes back to *G. Boole* and other famous mathematicians.

The other field is the use of the binary number system which has its roots in the papers of *G.F. Leibniz* and was *electrically implemented* by *C.E. Shannon* for relay circuits. It is nowadays a giant part of *information technology* and one of the fundamental concepts of circuit design.

The third part which only started in this time was the intention to contribute to the development of *artificial intelligence* by using rule-based systems. The implication played a major role in this field. We saw only later that our concepts will also cover this part. The developed algorithms were very efficient; it was possible to solve many problems in a very instructive way.

The last area was the *Boolean Differential Calculus* which existed at least in part at the end of the 1970s. Here it is tried to introduce and formalize the concept of changing the values of logic variables and functions and the emerging effects. In those days, it was only a research topic. At present, it is a huge area with a demanding theoretical background and many applications.

Therefore, the development of our research strategy was very time dependent. At the beginning, in the 1970s, we started with the development and the further clarification of the *Boolean Differential Calculus*. Based on the predecessor papers, we accepted a very long time ago the nomenclature of analysis; however, we were also aware of the fact that we explored finite algebraic structures. The development of the *concepts* of the Boolean differential calculus, their *extensions* to lattices of Boolean functions, and their *applications*, e.g., for combinational circuits, were the most important aspects of this work.

Everybody who wants to deal with parts of the developed theory must start with the elementary set $\mathbb{B} = \{0, 1\}$ and its properties. Who does not need all the single details of the algebraic structures can stay with the important basic properties. Thereafter, it is shown that the mentioned algebraic structures can be generalized to the set $\mathbb{B}^n$, the set of binary vectors with n components. A challenge is that n logic variables generate 2^n different binary vectors; most of the important problems have exponential complexity.

The key concept in the Boolean domain are logic functions, also called *Boolean functions*, *truth functions*, or *switching functions*. There are many different possibilities to represent a logic function. Starting with function tables the mapping to logic expressions, several types of normal forms and simplified forms are explained. For logic functions with few variables, the Karnaugh-map is introduced. Several types of decision trees can be used as a graphical representation of logic functions.

After the establishment of the theoretical foundations, we made an important step which has been based on the need to write programs for many problems. We were looking for an appropriate data structure and started (based on a paper of *A. Zakrevskij*) to use lists (matrices) of ternary vectors. This means mathematically that we transferred logic problems back to set theory. Each finite Boolean Algebra is equivalent to the power set of a finite set (with complement, intersection, union, difference, etc., as the relevant operations). An important algorithmic concept: many problems could be solved by working in parallel, on the level of bits and later also on the processor level. Therefore, the ternary vectors are the data structure that is used all the time; many search algorithms are more or less obsolete.

This shows also very impressively the cooperation between very different parts of Mathematics and Engineering with important contributions from both sides. The programming system XBOOLE was the result, in several levels and versions. The XBOOLE-Monitor can be used by everybody free of charge.

Very important: we saw that very different concepts of engineering could be expressed not only by functions, but by Boolean equations (and later by Boolean differential equations). Very soon we recognized that the XBOOLE system with the use of ternary vectors was an ideal tool for solving and teaching these problems. The use and the solution of all kinds of Boolean equations is the main tool "for everything."

Many concepts of the Boolean Differential Calculus for functions developed further and could also be successfully transferred to sets of Boolean functions, particularly to lattices of functions. Therefore, it was necessary and possible to include these results and important applications and make it available for a broader audience.

And here we found another field of interesting applications. Many problems of combinatorial nature, among them so-called NP- and *NP-complete problems*, could be represented and solved by means of Boolean equations. We were also able to replace in this field all kinds of search methods by directly solving the respective Boolean equations. Previously, when it could be seen that a problem has exponential complexity, then it was a good argument to put it aside. However, the highest complexity that was handled by our research had the value of $4^{324} \approx 10^{195}$. Therefore, the solution process of this problem and similar tasks is carefully presented, as an appeal to deal with the solution of these problems (if necessary).

The most important stimulus for the new edition of this book was the possibility to present the research work of about 40 years to others who can benefit from these results, theoretically as well as practically. We tried to write it in such a way that everybody should read the introductory part carefully; in the application part, however, he or she might be reading the chapter(s) he or she is interested in.

Hence, this book is not really a *revised edition*; it is a new edition that hopefully makes all the research results available for applications. It also aims to support the teaching starting from zero and reaching a level as high as necessary or possible.

Many people, colleagues, and students contributed to this book by their comments and discussions. We are grateful to all of them.

We are also very grateful to Springer who gave us the excellent possibility to transform the research work of many years into a textbook that is available for everybody and helpful for all the readers.

Chemnitz, Germany Christian Posthoff
Chemnitz, Germany Bernd Steinbach
July 2018

Contents

Acronyms

AF	Antivalence Form
ANF	Antivalence Normal Form
AP	Antivalence Polynomial
ASCII	American Standard Code for Information Interchange
BDD	Binary Decision Diagram
CF	Conjunctive Form
CMOS	Complementary Metal Oxide Semiconductor
CNF	Conjunctive Normal Form
DC	Don't Care
DEO	Derivative Operation
DF	Disjunctive Form
DNF	Disjunctive Normal Form
EF	Equivalence Form
ENF	Equivalence Normal Form
EP	Equivalence Polynomial
ESOP	Exclusive-OR Sum of Products
EXOR	Exclusive-OR
FSM	Finite-State Machine
GCD	Greatest Common Divisor
IS	Intermediate Solutions
ISF	Incompletely Specified Function
LCA	Logic Cell Array
LCM	Least Common Multiple
MCF	Most Complex Function
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
NTV	Number of Ternary Vectors
ODA	Orthogonal Disjunctive Form
OKE	Orthogonal Conjunctive Form
PLA	Programmable Logic Array
PLD	Programmable Logic Device
RAM	Random Access Memory

RGB	Red–Green–Blue
ROBDD	Reduced Ordered Binary Decision Diagram
ROM	Reed Only Memory
RT	Remaining Task
SA0-error	Stuck-at-zero error
SA1-error	Stuck-at-one error
SAT	Satisfiability
SAT-error	Stuck-at-T error
SNF	Special (or Specific) Normal Form
TM	Ternary Matrix
TVL	Ternary Vector List
UCP	Unate Covering Problem