London Mathematical Society Lecture Note Series 169

Boolean Function Complexity

Edited by
M. S. Paterson

CAMBRIDGE UNIVERSITY PRESS

This page intentionally left blank

LONDON MATHEMATICAL SOCIETY LECTURE NOTE SERIES

The books in the series listed below are available from booksellers, or, in case of difficulty, from Cambridge University Press.

34 Representation theory of Lie groups, M.F. ATIYAH *et al*
36 Homological group theory, C.T.C. WALL (ed)
39 Affine sets and affine groups, D.G. NORTHCOTT
40 Introduction to H_p spaces, P.J. KOOSIS
46 p-adic analysis: a short course on recent work, N. KOBLITZ
49 Finite geometries and designs, P. CAMERON, J.W.P. HIRSCHFELD & D.R. HUGHES (eds)
50 Commutator calculus and groups of homotopy classes, H.J. BAUES
57 Techniques of geometric topology, R.A. FENN
59 Applicable differential geometry, M. CRAMPIN & F.A.E. PIRANI
66 Several complex variables and complex manifolds II, M.J. FIELD
69 Representation theory, I.M. GELFAND *et al*
74 Symmetric designs: an algebraic approach, E.S. LANDER
76 Spectral theory of linear differential operators and comparison algebras, H.O. CORDES
77 Isolated singular points on complete intersections, E.J.N. LOOIJENGA
79 Probability, statistics and analysis, J.F.C. KINGMAN & G.E.H. REUTER (eds)
80 Introduction to the representation theory of compact and locally compact groups, A. ROBERT
81 Skew fields, P.K. DRAXL
82 Surveys in combinatorics, E.K. LLOYD (ed)
83 Homogeneous structures on Riemannian manifolds, F. TRICERRI & L. VANHECKE
86 Topological topics, I.M. JAMES (ed)
87 Surveys in set theory, A.R.D. MATHIAS (ed)
88 FPF ring theory, C. FAITH & S. PAGE
89 An F-space sampler, N.J. KALTON, N.T. PECK & J.W. ROBERTS
90 Polytopes and symmetry, S.A. ROBERTSON
91 Classgroups of group rings, M.J. TAYLOR
92 Representation of rings over skew fields, A.H. SCHOFIELD
93 Aspects of topology, I.M. JAMES & E.H. KRONHEIMER (eds)
94 Representations of general linear groups, G.D. JAMES
95 Low-dimensional topology 1982, R.A. FENN (ed)
96 Diophantine equations over function fields, R.C. MASON
97 Varieties of constructive mathematics, D.S. BRIDGES & F. RICHMAN
98 Localization in Noetherian rings, A.V. JATEGAONKAR
99 Methods of differential geometry in algebraic topology, M. KAROUBI & C. LERUSTE
100 Stopping time techniques for analysts and probabilists, L. EGGHE
101 Groups and geometry, ROGER C. LYNDON
103 Surveys in combinatorics 1985, I. ANDERSON (ed)
104 Elliptic structures on 3-manifolds, C.B. THOMAS
105 A local spectral theory for closed operators, I. ERDELYI & WANG SHENGWANG
106 Syzygies, E.G. EVANS & P. GRIFFITH
107 Compactification of Siegel moduli schemes, C-L. CHAI
108 Some topics in graph theory, H.P. YAP
109 Diophantine analysis, J. LOXTON & A. VAN DER POORTEN (eds)
110 An introduction to surreal numbers, H. GONSHOR
111 Analytical and geometric aspects of hyperbolic space, D.B.A. EPSTEIN (ed)
113 Lectures on the asymptotic theory of ideals, D. REES
114 Lectures on Bochner-Riesz means, K.M. DAVIS & Y-C. CHANG
115 An introduction to independence for analysts, H.G. DALES & W.H. WOODIN
116 Representations of algebras, P.J. WEBB (ed)
117 Homotopy theory, E. REES & J.D.S. JONES (eds)
118 Skew linear groups, M. SHIRVANI & B. WEHRFRITZ

119 Triangulated categories in the representation theory of finite-dimensional algebras, D. HAPPEL
121 Proceedings of *Groups - St Andrews 1985*, E. ROBERTSON & C. CAMPBELL (eds)
122 Non-classical continuum mechanics, R.J. KNOPS & A.A. LACEY (eds)
124 Lie groupoids and Lie algebroids in differential geometry, K. MACKENZIE
125 Commutator theory for congruence modular varieties, R. FREESE & R. MCKENZIE
126 Van der Corput's method of exponential sums, S.W. GRAHAM & G. KOLESNIK
127 New directions in dynamical systems, T.J. BEDFORD & J.W. SWIFT (eds)
128 Descriptive set theory and the structure of sets of uniqueness, A.S. KECHRIS & A. LOUVEAU
129 The subgroup structure of the finite classical groups, P.B. KLEIDMAN & M.W.LIEBECK
130 Model theory and modules, M. PREST
131 Algebraic, extremal & metric combinatorics, M-M. DEZA, P. FRANKL & I.G. ROSENBERG (eds)
132 Whitehead groups of finite groups, ROBERT OLIVER
133 Linear algebraic monoids, MOHAN S. PUTCHA
134 Number theory and dynamical systems, M. DODSON & J. VICKERS (eds)
135 Operator algebras and applications, 1, D. EVANS & M. TAKESAKI (eds)
136 Operator algebras and applications, 2, D. EVANS & M. TAKESAKI (eds)
137 Analysis at Urbana, I, E. BERKSON, T. PECK, & J. UHL (eds)
138 Analysis at Urbana, II, E. BERKSON, T. PECK, & J. UHL (eds)
139 Advances in homotopy theory, S. SALAMON, B. STEER & W. SUTHERLAND (eds)
140 Geometric aspects of Banach spaces, E.M. PEINADOR and A. RODES (eds)
141 Surveys in combinatorics 1989, J. SIEMONS (ed)
142 The geometry of jet bundles, D.J. SAUNDERS
143 The ergodic theory of discrete groups, PETER J. NICHOLLS
144 Introduction to uniform spaces, I.M. JAMES
145 Homological questions in local algebra, JAN R. STROOKER
146 Cohen-Macaulay modules over Cohen-Macaulay rings, Y. YOSHINO
147 Continuous and discrete modules, S.H. MOHAMED & B.J. MÜLLER
148 Helices and vector bundles, A.N. RUDAKOV et al
149 Solitons, nonlinear evolution equations and inverse scattering, M.J. ABLOWITZ & P.A. CLARKSON
150 Geometry of low-dimensional manifolds 1, S. DONALDSON & C.B. THOMAS (eds)
151 Geometry of low-dimensional manifolds 2, S. DONALDSON & C.B. THOMAS (eds)
152 Oligomorphic permutation groups, P. CAMERON
153 L-functions and arithmetic, J. COATES & M.J. TAYLOR (eds)
154 Number theory and cryptography, J. LOXTON (ed)
155 Classification theories of polarized varieties, TAKAO FUJITA
156 Twistors in mathematics and physics, T.N. BAILEY & R.J. BASTON (eds)
157 Analytic pro-p groups, J.D. DIXON, M.P.F. DU SAUTOY, A. MANN & D. SEGAL
158 Geometry of Banach spaces, P.F.X. MÜLLER & W. SCHACHERMAYER (eds)
159 Groups St Andrews 1989 volume 1, C.M. CAMPBELL & E.F. ROBERTSON (eds)
160 Groups St Andrews 1989 volume 2, C.M. CAMPBELL & E.F. ROBERTSON (eds)
161 Lectures on block theory, BURKHARD KÜLSHAMMER
162 Harmonic analysis and representation theory for groups acting on homogeneous trees, A. FIGA-TALAMANCA & C. NEBBIA
163 Topics in varieties of group representations, S.M. VOVSI
164 Quasi-symmetric designs, M.S. SHRIKANDE & S.S. SANE
165 Groups, combinatorics & geometry, M.W. LIEBECK & J. SAXL (eds)
166 Surveys in combinatorics, 1991, A.D. KEEDWELL (ed)
167 Stochastic analysis, M.T. BARLOW & N.H. BINGHAM (eds)
168 Representations of algebras, H. TACHIKAWA & S. BRENNER (eds)
169 Boolean function complexity, M.S. PATERSON (ed)
170 Manifolds with singularities and the Adams-Novikov spectral sequence, B. BOTVINNIK
173 Discrete groups and geometry, W.J. HARVEY & C. MACLACHLAN (eds)
174 Lectures on mechanics, J.E. MARSDEN
175 Adams memorial symposium on algebraic topology 1, N. RAY & G. WALKER (eds)
176 Adams memorial symposium on algebraic topology 2, N. RAY & G. WALKER (eds)
177 Applications of categories in computer science, M.P. FOURMAN, P.T. JOHNSTONE, & A.M. PITTS (eds)
178 Lower K- and L-theory, A. RANICKI
179 Complex projective geometry, G. ELLINGSRUD, C. PESKINE, G. SACCHIERO & S.A. STRØMME (eds)

London Mathematical Society Lecture Note Series. 169

Boolean Function Complexity

Edited by
M.S. Paterson
Department of Computer Science
University of Warwick

CAMBRIDGE
UNIVERSITY PRESS

CAMBRIDGE UNIVERSITY PRESS
Cambridge, New York, Melbourne, Madrid, Cape Town, Singapore, São Paulo

Cambridge University Press
The Edinburgh Building, Cambridge CB2 2RU, UK

Published in the United States of America by Cambridge University Press, New York

www.cambridge.org
Information on this title: www.cambridge.org/9780521408264

First published 1992

A catalogue record for this publication is available from the British Library

ISBN-13 978-0-521-40826-4 paperback
ISBN-10 0-521-40826-1 paperback

Transferred to digital printing 2006

Contents

Preface

Complexity theory attempts to understand and measure the intrinsic difficulty of computational tasks. The study of Boolean Function Complexity reaches for the combinatorial origins of these difficulties. The field was pioneered in the 1950's by Shannon, Lupanov and others, and has developed now into one of the most vigorous and challenging areas of theoretical computer science.

In July 1990, the London Mathematical Society sponsored a Symposium which brought to Durham University many of the leading researchers in the subject for ten days of lectures and discussions. This played an important part in stimulating new research directions since many of the participants were meeting each other for the first time. This book contains a selection of the work which was presented at the Symposium. The topics range broadly over the field, representing some of the differing strands of Boolean Function Theory.

I thank the authors for their efforts in preparing these papers, each of which has been carefully refereed to journal standards. The referees provided invaluable assistance in achieving accuracy and clarity. Nearly all the referees' names appear also in the list of authors, the others being A. Wigderson, C. Sturtivant, A. Yao and W. McColl. While a measure of visual conformity has been achieved (all but one of the papers is set using LaTeX), no attempt was made to achieve uniform notation or a 'house style'. I have tried to arrange the papers so that those which provide more introductory material may serve to prepare the reader for some more austere papers which follow. Some background in Boolean complexity is assumed for most of the papers. A general introduction is offered by the three books by Dunne, Savage and Wegener which are referenced in the first paper.

The Symposium at Durham was made possible by the initiative and sponsorship of the London Mathematical Society, the industry and smooth organization of the staff at Durham University, the financial support of the Science and Engineering Research Council and by the enthusiastic participation of the Symposium members. Finally, I thank the staff and Syndics of Cambridge University Press for their cooperation and patience during the preparation of this volume.

Mike Paterson

University of Warwick

Coventry, England

June, 1992

Participants listed from left to right.

Top row: R. Raz, K. Edwards, N. Nisan, L. Valiant, A. Macintyre, K. Kalorkoti, W. Beynon, R. Smolensky, I. Newman, D. Uhlig, A. Chin, I. Leader, U. Zwick, C. Sturtivant, G. Brightwell.

Middle row: M. Jerrum, A. Cohen, A. Sinclair, A. Borodin, C. Schnorr, A. Wilkie, A. Andreev, N. Biggs, P. d'Aquino, M. Dyer, P. Dunne, A. Thomason.

Bottom row: I. Wegener, A. Stibbard, N. Pippenger, M. Klawe, J. Savage, M. Furst, A. Widgerson, W. McColl, M. Paterson, A. Yao, A. Razborov, M. Sipser, L. Henderson, C. McDiarmid, J. Shawe-Taylor, D. Barrington, R. Mirwald.

Participants in the
LMS symposium on Boolean Function Complexity,
Durham University, July 1990.

Miklos Ajtai
Alexander Andreev
David Mix Barrington
Alessandro Berarducci
Meurig Beynon
Norman Biggs
Béla Bollobás
Allan Borodin
Graham Brightwell
Andrew Chin
Aviad Cohen
Paola d'Aquino
Paul Dunne
Martin Dyer
Keith Edwards
Merrick Furst
Leslie Henderson
Mark Jerrum
Kyriakos Kalorkoti
Maria Klawe
Imre Leader
Angus Macintyre
William McColl
Colin McDiarmid
Roland Mirwald
Ilan Newman
Noam Nisan
Margarita Otero
Mike Paterson
Nicholas Pippenger
Franco Preparata
Ran Raz
Alexander Razborov
John Savage
Claus Schnorr
John Shawe-Taylor
Alistair Sinclair
Mike Sipser
Roman Smolensky
Alyson Stibbard
Carl Sturtivant
Andrew Thomason
Dietmar Uhlig
Leslie Valiant
Ingo Wegener
Avi Wigderson
Alex Wilkie
Andrew Yao
Uri Zwick

Relationships Between Monotone and Non-Monotone Network Complexity

Paul E. Dunne *

Abstract

Monotone networks have been the most widely studied class of restricted Boolean networks. It is now possible to prove superlinear (in fact exponential) lower bounds on the size of optimal monotone networks computing some naturally arising functions. There remains, however, the problem of obtaining similar results on the size of combinational (i.e. unrestricted) Boolean networks. One approach to solving this problem would be to look for circumstances in which large lower bounds on the complexity of monotone networks would provide corresponding bounds on the size of combinational networks.

In this paper we briefly review the current state of results on Boolean function complexity and examine the progress that has been made in relating monotone and combinational network complexity.

1. Introduction

One of the major problems in computational complexity theory is to develop techniques by which non-trivial lower bounds, on the amount of time needed to solve 'explicitly defined' decision problems, could be proved. By 'non-trivial' we mean bounds which are superlinear in the length of the input; and, since we may concentrate on functions with a binary input alphabet, the term 'explicitly defined' may be taken to mean functions for which the values on all inputs of length n can be enumerated in time 2^{cn} for some constant c.

* Department of Computer Science, University of Liverpool, Liverpool, L69 3BX, Great Britain.

Classical computational complexity theory measures 'time' as the number of moves made by a (multi-tape) deterministic Turing machine. Thus a decision problem, f, has *time complexity*, $T(n)$ if there is a Turing machine program that computes f and makes at most $T(n)$ moves on any input of length n.

The Turing machine is only one of many different models of computation. Another model, that has attracted as much attention, is the class of *combinational Boolean networks.* An *n-input combinational network* is a directed acyclic graph containing two distinct types of node: *input nodes*, which have no incoming edges; and *gate* nodes which have at most two incoming edges. Each input node is associated with a single Boolean variable, x_i, from an ordered set $\mathbf{X_n} = \langle x_1, x_2, \ldots, x_n \rangle$. Each gate node is associated with some two-input Boolean function. There is a unique gate, having no outgoing edges, which is called the *output* of the network. An assignment of Boolean values to the input variables naturally induces a Boolean value at the output gate, the actual value appearing depends on the input assignment and the network structure. The *size* of such a network is the number of gate nodes; its *depth* is the number of gates in the longest path from an input node to the output gate.

We shall denote by B_n the set of all n-input Boolean functions, $f(\mathbf{X_n})\{0,1\}^n \rightarrow \{0,1\}$ with formal arguments $\mathbf{X_n}$. An n-input combinational network computes $f \in B_n$ if for all assignments $\alpha \in \{0,1\}^n$ to $\mathbf{X_n}$, the value induced at the output gate is $f(\alpha)$. It should be noted that a single combinational network only solves a decision problem for the special case when all input strings are of length exactly n. In order to discuss the size (or *combinational complexity*) of networks for decision problems in general, the following approach is used. Let $[f_n]$ be the infinite sequence of Boolean functions arising by restricting a decision problem, f, to inputs of length n (thus $f_n \in B_n$). We say that the decision problem, f, is computed by a sequence of n-input combinational networks, $\langle C_n \rangle$, if, for each n, the n-input network, C_n, computes f_n. With this definition we can introduce appropriate complexity measures for Boolean functions computed by networks.

For a network, T, $\mathbf{C}(T)$ is the size of T; for a Boolean function $f \in B_n$

$$\mathbf{C}(f) = \min\{\mathbf{C}(T) : T \textit{ computes } f\}$$

Finally for a family $[f_n]$ we say that the combinational complexity of $[f_n]$ is $g(n)$ if, for each f_n, it holds that $\mathbf{C}(f_n) \leq g(n)$. $\mathbf{D}(f)$ will denote the corresponding measure for depth.

If a decision problem can be computed in time $T(n)$ then $T(n)\log T(n)$ is an upper bound on the combinational complexity of the corresponding family of Boolean functions, see, e.g. Savage (1972), Schnorr

(1976a) or Fischer and Pippenger (1979). In this way sufficiently large lower bounds on combinational complexity would give similar bounds on Turing machine time. Lower bounds on Turing machine space could be obtained from $\omega(\log^2 n)$ lower bounds on combinational depth, cf. Borodin (1977).

In fact it is known that there are Boolean functions of n-arguments with exponential combinational complexity. Shannon (1949) proved that 'almost all'[1] $f \in B_n$ were such that $\mathbf{C}(f) \geq 2^n/n$. Earlier, Riordan and Shannon (1942) had proved that, for almost all $f \in B_n$, $\mathbf{D}(f) \geq n - \log\log n$. Lupanov (1958) (for size) and Gaskov (1978) (for depth) have established that these lower bounds are the best possible and so a lot is known about the difficulty of computing Boolean functions, by combinational networks, in the general case.

If we consider the case of explicitly defined Boolean functions, however, the existing results are extremely weak. To date, no superlinear lower bound has been proved on the combinational complexity of any specific function: the largest lower bound proved, is only $3n-3$ for a function constructed in Blum (1984a). It has become clear that, if combinational networks are to provide a vehicle with which to derive superlinear lower bounds on Turing machine time — let alone resolve questions such as $P = ?\,NP$ — then techniques that are much more sophisticated, than those developed to date, must be constructed. In the absence of such methods, attention has been focused on restricted types of combinational networks. There are a number of reasons for proceeding along this path: one cannot hope to prove results on unrestricted networks unless one can prove results for special cases; understanding how to prove lower bounds on restricted types of network may give some insight into techniques that can be applied to the general case; and it may be possible to deduce lower bounds on combinational complexity from lower bounds on restricted networks, for example if the special class of networks can efficiently simulate combinational networks.

In this paper we are concerned with a particular class of restricted combinational network: monotone Boolean networks. These are introduced in Section 2, where a survey of lower bound results obtained for this model is also given. The remainder of the paper deals with the issue of relating monotone network complexity to combinational complexity: Section 3 describes a framework for translating between combinational and monotone networks and, within this, a class of functions known as *slice functions* may be shown to have closely related combinational and monotone network complexity. Slice functions and their properties are examined, in detail, in Section 4.

1) A property holds for 'almost all' $f \in B_n$ if the fraction of all n-input Boolean functions not possessing the property approaches zero as n approaches infinity.

Conclusions are given in the final section. The reader interested in progress on other aspects of combinational complexity or alternative restricted models may find discussions of work in these areas in Dunne (1988), Savage (1976), and Wegener (1987).

2. Monotone Boolean Networks

Combinational networks allow any two-input Boolean function to be used as a gate operation. The restriction imposed in the case of monotone Boolean networks is that the only gate operations admitted are two-input logical AND (or *conjunction*) — denoted $\wedge$ — and two-input logical OR (or *disjunction*) — denoted $\vee$. For Boolean variables x, y: $x \wedge y$ equals 1 if and only if both x and y equal 1; $x \vee y$ equals 1 if and only if at least one of x or y equals 1.

There is a penalty incurred by imposing this restriction on networks: it is no longer possible to compute every Boolean function of n arguments. In other words, the *basis* (i.e. permitted set of operations) $\{\wedge, \vee\}$ is *logically incomplete*. Post (1941) described necessary and sufficient conditions for a basis to be logically complete. In the next section we exploit two facts about complete bases, namely:

Fact 2.1: The basis $\{\wedge, \vee, \neg\}$ (where $\neg$ is the unary function corresponding to Boolean negation) is logically complete. □

Fact 2.2: If $\Omega \subseteq B_2$ is a complete basis then the size of an optimal Boolean network, using only operations in Ω, computing a function $f \in B_n$ is at most $c\,\mathrm{C}(f)$ for some (small) constant c. □

A function which can be computed by a monotone Boolean network is called a *monotone Boolean function.* M_n denotes the (strict) subset of B_n comprising all n-input monotone Boolean functions. The study of this class of functions dates back to the work of Dedekind (1897) where the problem of calculating the exact value of $\psi(n) = |M_n|$ was first raised. This exact counting problem is still open, although asymptotically exact estimates have been obtained, cf. Korshunov (1981).

Monotone Boolean functions have a number of interesting properties which have proved important in constructing lower bound arguments for monotone network complexity. A few of these properties are summarised below.

Before stating these we need the following concepts. Define ordering relations $\leq$ and $<$ on Boolean functions as follows: $0 < 1$ and for f, g in B_n we say that $f \leq g$ if for all $\alpha \in \{0,1\}^n$, $f(\alpha) = 1 \Rightarrow g(\alpha) = 1$. That is, whenever some assignment makes f take the value 1, the same assignment forces g to take the value 1. We say that $f < g$ if $f \leq g$ but f and g are different functions. Now let f and g be functions in B_n with formal arguments $\mathbf{X_n}$. $f^{|x_i:=\varepsilon}$ denotes the function (in B_{n-1} with formal arguments $\mathbf{X_n}-\{x_i\}$) obtained by fixing x_i to the Boolean value ε.

Fact 2.3: Let $f \in B_n$ and let $\mathbf{X_n}$ be the formal arguments of f. $f \in M_n$ if and only if: $\forall x_i$, $1 \le i \le n$ it holds that $f^{|x_i:=0} \le f^{|x_i:=1}$. □

Fact 2.4: If f, g are in M_n and $f \le g$ then:

i) $f \wedge g = f$

ii) $f \vee g = g$. □

A conjunction of some subset of the variables $\mathbf{X_n}$ is called a *monom*. A monom, m, is an *implicant* of $f \in M_n$ if $m \le f$. A monom, m, is a *prime implicant* of f if m is an implicant of f but no monom formed from a strict subset of the variables of m is an implicant of f. $\mathbf{PI}(f)$ will denote the set of prime implicants of f. The dual concepts, using disjunction, are clauses, implicands, and prime clauses with $\mathbf{PC}(f)$ denoting the set of prime clauses of a function f.

Fact 2.5: Any $f \in M_n$, with arguments $\mathbf{X_n}$, may be expressed uniquely in the forms

$$f(\mathbf{X_n}) = \bigvee_{p \in \mathbf{PI}(f)} p \quad ; \quad f(\mathbf{X_n}) = \bigwedge_{q \in \mathbf{PC}(f)} q$$

The former is known as *Disjunctive Normal Form* (DNF); the latter as *Conjunctive Normal Form* (CNF). □

$\mathbf{C^m}(f)$ will denote the monotone network complexity of $f \in M_n$ and $\mathbf{D^m}(f)$ the corresponding measure for monotone depth.

Early progress on the complexity of monotone Boolean networks was similar to the case of combinational networks. Thus there are asymptotically exact bounds for the monotone network size of almost all monotone Boolean functions. The lower bound (of $2^n/n^{3/2}$) follows from Gilbert (1954) using Shannon's arguments; the upper bound comes from Andreev (1988) (improving the constant factor in the construction of Red'kin (1979)).

The first significant development in the theory of monotone networks came about with the appearance of superlinear lower bounds on the size of monotone networks computing *sets* of monotone Boolean functions: 'superlinear' in this context means as a function of the total number of inputs and outputs. Van Voorhis (1972) proved an asymptotically optimal lower bound on the monotone network complexity of sorting n Boolean inputs; Paterson (1975) and Mehlhorn and Galil (1976) independently obtained exact bounds on the size of networks realising $(\wedge, \vee)$-Boolean matrix product; Weiss (1984) and Blum (1984b) obtained lower bounds for the n-point Boolean convolution function which is closely related to integer multiplication.

In the case of single monotone Boolean functions, until recently, as little progress had been made as for combinational networks. Although exact exponential lower bounds had been obtained by Schnorr (1976b) and Jerrum and Snir (1982) for monotone *arithmetic* networks (i.e. with only integer addition and multiplication permitted as operations) the techniques used to prove these results fail to work for algebraic structures in which the identities of Fact 2.4 hold. By the end of 1984 the most powerful techniques were capable of yielding only modest linear lower bounds, e.g. Dunne (1985), Tiekenheinrich (1984).

In 1985 the Soviet mathematician Razborov considered the following monotone Boolean functions.

Definition 2.1: Let $\mathbf{X_n^U} = \{ x_{i,j} : 1 \le i < j \le n \}$ be a set of $N = n(n-1)/2$ Boolean variables representing the adjacency matrix of an n-vertex undirected graph $G(\mathbf{X_n^U})$. *k-clique* is the function in M_N, with formal arguments $\mathbf{X_n^U}$, such that *k-clique*$(\alpha) = 1$ if the graph $G(\alpha)$ contains a k-clique, i.e. a set of k vertices every pair of which is joined by an edge of G.

Let $\mathbf{X_{n,n}} = \{ x_{i,j} : 1 \le i, j \le n \}$ be a set of n^2 Boolean variables. The *Logical Permanent* is the function $PM \in M_{n^2}$, with formal arguments $\mathbf{X_{n,n}}$, defined by

$$PM(\mathbf{X_{n,n}}) = \bigvee_{\sigma \in S_n} \bigwedge_{i=1}^{n} x_{i,\sigma(i)}$$

where S_n is the set of all permutations of $\langle 1, 2, \ldots, n \rangle$. •

For appropriate (non-constant) values, the decision problem corresponding to the k-clique function is NP-complete.

Alon and Boppana (1986), improving the combinatorial arguments given originally in Razborov (1985a, 1985b), proved the following results concerning these functions.

Theorem 2.1: $\forall\ 3 \le k < 0.25\,(n/\log n)^{2/3}$

$$\mathbf{C^m}(k-clique) \ge c \left(\frac{n}{16\,k^{3/2} \log n} \right)^{\sqrt{k}} \qquad \square$$

Theorem 2.2:

$$\mathbf{C^m}(PM) \ge n^{c \log n} \qquad (\forall\ c < 1/16) \qquad \square$$

The lower bound of Theorem 2.1 is exponential for large enough values of k. In addition to these results of Razborov, Alon, and Boppana, exponential lower bounds on explicitly defined monotone Boolean functions have been proved in Andreev (1985, 1987) and Tardos (1987).

Theorems 2.1 and 2.2 constitute a significant advance in the theory of Boolean network complexity since they are built on a technique which is powerful enough to yield superlinear lower bounds on size for a non-trivial network model. Further indications that monotone networks are a theoretically tractable model are given by the methods of Karchmer and Wigderson (1987) and Raz and Wigderson (1990). Their results concern the depth of monotone networks.

Definition 2.2: The function *st–conn*($\mathbf{X}_n^U$) is the monotone Boolean function such that *st–conn*(α)=1 if $G(\alpha)$ contains a path from vertex s to vertex t.

Theorem 2.3: (Karchmer and Wigderson, 1987)

$$\mathbf{D}^m(st-conn) = \Omega(\log^2 n) \qquad \square$$

Theorem 2.4: (Raz and Wigderson, 1990)

$$\mathbf{D}^m(PM) = \Omega(n) \qquad \square$$

Razborov (1988) also proves superlogarithmic lower bounds on monotone depth.

3. A Framework for Relating Combinational and Monotone Network Complexity

The theorems stated at the conclusion of the preceding section may be regarded as completing the first part of a programme aimed at achieving non-trivial lower bounds on problem complexity. Thus, for the restricted case of monotone networks, techniques powerful enough to prove large lower bounds on size and depth are known. The question that now arises is: how relevant are these results/techniques to combinational complexity? In other words: is it possible to deduce non-trivial lower bounds on combinational complexity (depth) from large enough lower bounds on monotone complexity (depth)?

The results of Razborov (1985b), Tardos (1987) and Raz and Wigderson (1990), at first sight, offer a negative answer to the second question.

Theorem 3.1:

i) $\mathbf{C}(PM) = O(n^k)$ for some constant k.

ii) There is function computable with polynomial size combinational networks that requires exponential size monotone networks.

iii) There is a function computable in $O(\log n)$ depth using combinational networks that requires $\Omega(\sqrt{n})$ depth monotone networks.

Proof: (i) follows by observing that the Logical Permanent is equivalent to determining whether a given bipartite graph contains a perfect matching. Hopcroft and Karp (1973) give a polynomial time algorithm for this problem and thus the upper bound on combinational complexity is immediate. (ii) is proved in Tardos (1988) and (iii) by Raz and Wigderson (1990). □

The second and third parts of Theorem 3.1 (which are both proved using explicitly defined functions) show that there are exponential gaps between monotone network size (depth) and combinational network size (depth). As a consequence it will not *always* be possible to derive lower bounds on combinational complexity using lower bounds on monotone complexity. Nevertheless the theorem does not exclude the possibility of doing this for *some* monotone Boolean functions.

Recall from Facts 2.1 and 2.2 that the basis $\{\wedge,\vee,\neg\}$ is logically complete and that an optimal Boolean network built from any complete basis of two-input Boolean operations is at most a constant factor larger than an equivalent optimal combinational network. It follows that, since we are interested in superlinear lower bounds, we may without loss of generality consider the problem of relating monotone networks to networks which only permit the operations $\{\wedge,\vee,\neg\}$ to be used.

$\{\wedge,\vee,\neg\}$-networks only differ from monotone networks in permitting the use of negation. The result below demonstrates that we can make such networks more closely resemble monotone networks by permitting the use of negation only on *input* nodes. We shall use $C_{\{\wedge,\vee,\neg\}}(f)$ to denote the number of gates in the smallest $\{\wedge,\vee,\neg\}$-network realising $f \in B_n$.

Definition 3.1: A *standard network* is a Boolean network whose permitted gate operations are $\{\wedge,\vee\}$ and with $2n$-input nodes:

$$\langle x_1,\ldots,x_n,\neg x_1,\ldots,\neg x_n\rangle$$

SC(f) will denote the number of *gate* nodes in the smallest standard network realising $f \in B_n$. •

Theorem 3.1: $\forall f \in B_n$ it holds that $SC(f) \leq 2\,C_{\{\wedge,\vee,\neg\}}(f)$.

Proof: (Outline) The following identities (known as De Morgan's Laws) can be easily proved:

$$\neg(x \wedge y) = (\neg x) \vee (\neg y) \quad ; \quad \neg(x \vee y) = (\neg x) \wedge (\neg y)$$

Let T be an optimal $\{\wedge,\vee,\neg\}$-network realising some $f \in B_n$. Let g be a 'last' gate in T such that an edge directed out of g enters a negation gate. Here 'last' means that no gate on a path from g to the output gate has the property that an edge directed out of it enters a negation gate. Now since we include instances of negation in measuring size and we have assumed that T is optimal it follows that there is *exactly one* wire leaving g and entering a negation gate, h say. Let $h_1,\ldots,h_r$ be the gates which have h as an input. Let g_1 and g_2 be the gates supplying the inputs of g. We change T as follows: add a new gate g' whose inputs are $\neg g_1$ and $\neg g_2$; remove the negation gate h and replace each edge $\langle h, h_i \rangle$ by an edge $\langle g', h_i \rangle$; finally if g is an $\wedge$-gate then make g' an $\vee$-gate and vice versa. From De Morgan's Laws it follows that the new network, T', still computes f.

Applying the process of the preceding paragraph repeatedly, we eventually reach the situation where only input nodes enter a negation gate. Since we add only one new ($\wedge$ or $\vee$) gate at each stage it follows that the final network is a standard network computing f and containing at most twice the number of gates in T. □

Now consider an optimal combinational network, T, computing some $f \in M_n$. This may be transformed to a standard network, S, that also computes f, and is only a constant factor larger than T. The only way in which S differs from a monotone network is by the presence of the n extra input nodes $\langle \neg x_1, \ldots, \neg x_n \rangle$.

Suppose that we, temporarily, ignore the fact that the n additional inputs are the negation of the n function arguments and regard them as n new Boolean variables $y_1, \ldots, y_n$. Then it is clear that:

i) S computes a *monotone* Boolean function of the inputs $\langle x_1, \ldots, x_n, y_1, \ldots, y_n \rangle$.

ii) If, for each i, we substitute $\neg x_i$ for the input y_i then S computes the original function $f \in M_n$.

One of the most important techniques applied in proving lower bounds on monotone network complexity is the concept of *replacement rules*. These prescribe 'circumstances' in which a node of a monotone network computing some function $h(\mathbf{X_n})$ may be *replaced* by a node computing some *different* function $h'(\mathbf{X_n})$ without altering the function, f, computed by the network. The 'circumstances' depend solely on h, h' and f and *not* on the topology of the network.[2]

Returning to the standard network S in which $\neg x_i$ is regarded as a new input y_i we can attempt to use the concept of replacement rules to yield a monotone network with inputs $\mathbf{X_n}$ which computes f. Thus, if the following two conditions can be satisfied, *for all* standard networks computing f, we may deduce that $\mathbf{C^m}(f)$ is 'not much larger' than $\mathbf{C}(f)$.

C1) There is a set $\langle h_1, \ldots, h_n \rangle$ of monotone Boolean functions having formal arguments $\mathbf{X_n}$ such that replacing any subset of the y_i inputs by the

2) The power of this technique arises from the fact that one may identify functions which can be replaced by the Boolean constants 0 or 1 and thus cannot be computed as partial results in optimal monotone networks. An example of the technique in practice may be found in Paterson (1975). A full characterisation of applicable replacements is given in Dunne (1984, 1988), see also Beynon's paper in this volume.

corresponding h_i functions and the remaining y_j inputs by the corresponding $\neg x_j$ inputs, results in a network computing f.

C2) The set of n monotone Boolean functions $\langle h_1, \ldots, h_n \rangle$ can be computed by a *monotone* network of size at most $\varepsilon_n \mathbf{C}^{\mathbf{m}}(f)$ (for some $\varepsilon_n < 1$).

Theorem 3.3: If $f \in M_n$ for which conditions (C1) and (C2) hold, then

$$\mathbf{C}(f) \geq \frac{1-\varepsilon_n}{2c} \mathbf{C}^{\mathbf{m}}(f)$$

where c is the constant of Fact 2.2.

Proof: If both (C1) and (C2) hold then it follows that $\mathbf{C}^{\mathbf{m}}(f) \leq \mathbf{SC}(f) + \varepsilon_n \mathbf{C}^{\mathbf{m}}(f)$. The theorem now follows from Fact 2.2 and Theorem 3.2. □

For $f \in M_n$, a set $\langle h_1, \ldots, h_n \rangle$ of monotone functions satisfying condition (C1) for f, is called a *pseudo-complement vector for f*. h_i is called a *pseudo-complement* for x_i when computing f. Informally a pseudo-complement for x_i can replace the node $\neg x_i$ in *any* standard network computing f.

Given the relation in Theorem 3.3, it is clearly desirable to identify classes of monotone Boolean functions for which both conditions (C1) and (C2) hold. In fact it turns out that (C1) holds *for all* $f \in M_n$.

Theorem 3.4: $h \in M_{n-1}$ with formal arguments $\mathbf{X_n} - \{x_i\}$ is a pseudo-complement for x_i when computing $f \in M_n$ (with arguments $\mathbf{X_n}$) if and only if

$$f^{|x_i:=0} \leq h \leq f^{|x_i:=1}$$

Proof: The result was originally proved in Dunne (1984). This proof is reproduced in Dunne (1988) pp. 242-243. □

Corollary 3.1: $\forall\ f \in M_n$ condition (C1) holds.

Proof: From Fact 2.3, $f \in M_n$ if and only if $f^{|x_i:=0} \leq f^{|x_i:=1}$ for each x_i. It follows that the interval of Theorem 3.4 is always well-defined. □

Theorem 3.4 does not, however, allow functions for which condition (C2) holds to be identified directly. An 'obvious' choice of pseudo-complement vector, such as the n subfunctions of f obtained by fixing x_i to 0, will not give an efficient transformation from standard networks to monotone networks. Theorem 3.4 is mainly of use in permitting simple proofs of the *correctness* of specific pseudo-complements.

Rather than attempt to identify, explicitly, those $f \in M_n$ for which (C2) holds, i.e. for which efficiently computable pseudo-complement vectors exist, we proceed in the 'reverse direction'. Thus: