

Contents

Preface	i
1) Introduction	1
1.1) Complexity Theory	1
1.2) Boolean functions and Boolean algebra	7
1.3) Boolean networks	18
Bibliographic notes	25
2) Combinational Network Complexity	27
2.1) Simulation results	28
2.2) Complexity results for almost all Boolean functions	42
2.3) Relating network size and network depth	68
2.4) Lower bounds on specific Boolean functions	74
2.5) Some upper bounds on combinational complexity	100
Bibliographic notes	115
3) Monotone Network Complexity	117

3.1) Bounds for almost all monotone Boolean functions	122
3.2) Replacement rules	147
3.3) The monotone complexity of sets of functions	154
3.4) Linear lower bounds on single output functions	172
3.5) Superpolynomial bounds on single output functions	195
3.5.1) The lattice method	196
3.5.2) The Andreev lower bound method	224
3.5.3) Conclusion	232
3.6) Relating monotone and combinational complexity	238
3.6.1) Standard circuits and pseudo – complementation	239
3.6.2) Slice functions	243
Bibliographic notes	269
4) Formulae	271
4.1) Bounds on formula size for all Boolean functions	272
4.2) General lower bound techniques	280

4.2. 1) The Neciporuk bound	281
4.2. 2) The Hodes/Specker/Pudlak lower bound	293
4.2. 3) The Fischer/Meyer/Paterson lower bound	303
4.3) Formula size and depth	322
4.4) Upper bounds for symmetric functions	330
4.5) Bounds for bases other than B_2	343
4.5. 1) The Khrapchenko bound	344
4.5. 2) The Andreev bound	347
Bibliographic notes	351
5) Bounded – Depth Networks	353
5.1) Introduction	353
5.2) Universal bounds on bounded – depth formulae	357
5.3) Exponential lower bounds on parity functions	364
5.4) Consequences of the parity function lower bound	372
5.5) Bounded – depth $\{\wedge, \oplus\}$ – formulae	382

Bibliographic notes	394
6) Planar Networks	396
6.1) Introduction	396
6.2) Relations between planar and combinational complexity	397
6.3) Bounds on planar network complexity	405
6.4) Planar networks and VLSI circuits	413
Bibliographic notes	420
Bibliography	422

List Of Figures

Figure 1. 1	3		
Figure 2. 1	38	Figure 2. 7	99
Figure 2. 2	82	Figure 2. 8 (a)	102
Figure 2. 3	79	Figure 2. 8 (b)	103
Figure 2. 4	86	Figure 2. 9	108
Figure 2. 5	87	Figure 2. 10	109
Figure 2. 6	88	Figure 2. 11	114
Figure 3. 1	137	Figure 3. 6	180
Figure 3. 2	175	Figure 3. 7	181
Figure 3. 3	176	Figure 3. 8	184
Figure 3. 4	178	Figure 3. 9	241
Figure 3. 5	179		
Figure 6. 1	400	Figure 6. 4	403
Figure 6. 2	401	Figure 6. 5	406

Figure 6.3 402

Figure 6.6

411

Preface

Quam multa fieri non posse, priusquam sunt facta, judicantur ?

Pliny The Elder

Historia Naturalis, VII, 1

From the early work of George Boole on the algebra of logic through to the present, the study of discrete, particularly Boolean, functions and their realisation has been actively pursued by numerous researchers. Yet, despite the pioneering investigations of Shannon, Riordan and others in the 1940s, this subject was largely neglected as a separate mathematical discipline for over two decades outside the Soviet Union; there the distinguished contribution of one generation of theoreticians laid the foundations of Boolean complexity theory. It is only in the last 15 years that this study has come to be widely recognised as a fundamental concern within the realm of modern computational complexity theory.

Since the appearance of Savage (1976), the first general survey of Boolean function complexity, there have been enormous advances in this theory coupled with the growth of new topics. Two areas in particular serve to illustrate these developments: the study of bounded-depth networks, which has led to important results in the domain of structural complexity; and the theory of monotone networks, which below occupies a significant portion of the text. The abundance of

new results in the latter field, most notably the work of Razborov and Berkowitz, has strengthened the argument that a new book taking account of these and other advances is required.

My aim in writing this monograph was originally to give a comprehensive survey of all major results, relevant to the sphere of Boolean network complexity, up to the end of 1987. Unfortunately considerations of space have made it impossible to include detailed presentations of every pertinent topic. It has thus been necessary to omit a number of subjects either on the grounds that they no longer have the significance they once did, e.g synchronous combinational complexity; or because they could not be treated adequately in the context of a general work. The main victim of the second policy is the growing field of uniform circuit complexity; this, I believe, is substantial enough to merit a book to itself.

As far as possible I have tried to make the material self-contained, however some familiarity with discrete mathematical topics; sets, relations, probability theory and combinatorics, is assumed.

The opening rhetorical question is quoted not out of immodest pride on having completed the text, rather out of a sense of relief that 30 months work is finished; that this occasion should arise often seemed, while writing, to be one of the *multa fieri non posse judicantur* ! All too often is it left unsaid that a book of this nature is the work of the author only in the narrow sense that he chooses the words and is responsible for the errors. It is therefore a pleasure to thank the many individuals who have helped me in the preparation of the text; in particular Mike Paterson, for invaluable comments on the first drafts and for contributing a number of new proofs of existing results; Sacha Razborov, for pointing out some misconceptions in the original draft of Chapter(3); Ingo Wegener, who generously supplied a copy of his own (excellent) recent volume covering similar topics;

Stassys Jukna who kept me informed of and provided several, otherwise unobtainable, significant papers which have appeared recently in Soviet technical journals; first Rosemary Altoft and later Andrew Carrick of Academic Press for tolerating the slow progress of the manuscript and the frequently revised "final" dates of completion; and finally my colleagues in the Computer Science department of Liverpool University for many thought provoking discussions on text-formatting, complexity theory and other matters too numerous to mention.

P.E.D

Liverpool, June 1988

... the reader who has got as far as the preface and been discouraged by that, has spent money on the book; he wishes to know how he is to be compensated for his expenditure. I can only remind him that he knows of several ways of using a book without actually reading it: he can use it to fill a space in his library where, neatly bound, it is sure to look good; or he may leave it lying upon a table to be seen by erudite friends. Or finally he can review it; this must surely be the best option ...

Arthur Schopenhauer

**Preface to the 1st Edition of
Die Welt als Wille und Vorstellung**

Chapter 1

Introduction

...they had built their hope of heaven on the binary system and the computer, 1 and 0, Yes and No...

Norman Mailer

The Armies of the Night

1.1) Complexity Theory

In very broad terms complexity theory is that field of Computer Science concerned with formally reasoning about how "difficult" specific problems are to solve. In order to make this rather general description more precise one must consider the following questions.

- Q1) How is a "problem" specified?
- Q2) What does a "solution" consist of?
- Q3) How is "difficulty" being measured?

For our purposes any problem can be viewed as associating a particular result or output with each valid input, that is as a function (f) from some domain of input values (I) to some range of output values (O). In this book we are exclusively concerned with functions for which both input and output values are encoded as finite strings of Boolean/binary values i.e 0 (or False) and 1 (or True). The functions where the result is a single Boolean value will in this section be referred to as *decision problems*. These correspond to

problems which ask whether some property is true of the input, e.g is a positive integer, encoded in binary, a prime number. The *size* of a problem instance is defined to be the length of its input. The class of functions introduced above will be considered in greater detail in Section(1.2).

With this approach "solving a problem" is equivalent to computing some function f , and by a solution or computation we mean a precisely specified sequence of instructions which given some input string \mathbf{x} returns the result \mathbf{y} such that $f = \mathbf{y}$. Such a sequence is called an *algorithm* for f .

Computation is considered as performed on abstract machines or *models of computation* which encapsulate intuitive notions of computer operation without reference to any specific realised architecture. Undoubtedly the best known of such models is that proposed by Turing (1936). The definition below differs from it only in minor technical details.

Definition 1.1: A k -tape Turing Machine (TM) is defined by a 7-tuple:

$$M = (Q, \Gamma, B, q_0, \delta, q_A, q_R)$$

where:

Q is a finite set of states.

Γ is a finite alphabet of tape symbols (we shall assume $\Gamma = \{0, 1\}$)

$B \notin \Gamma$ is the blank symbol.

$q_0 \in Q$ is the initial state.

$\delta: Q \times \{\Gamma \cup \{B\}\}^{k+1} \rightarrow Q \times \{\Gamma\}^k \times \{L, R, S\}^{k+1}$ is the state transition function.

$q_A, q_R \in Q$ are final states. q_A is the *accept* state and q_R the *reject* state.

M has an *input tape* and k *work-tapes*. Each tape is divided into infinitely many cells numbered $\dots, -2, -1, 0, 1, 2, \dots$. A cell can record precisely one symbol at a time. Each work-tape is scanned by a two-way read-write *head*, and the input tape by a two-way read-only head. The operation of M is supervised by a *finite control*. (Figure 1.1) •

Figure 1.1

M solves a decision problem as follows.

Initially all tape heads are positioned at cell 0 on their corresponding tapes. Cells $0..n-1$ of the input tape contain the input data, \mathbf{x} and all other cells contain the blank symbol. The initial state is q_0 .

In a single computation step (or *move*) the following actions are performed. M reads the symbol at the head position on each tape. The

current state and the $k+1$ symbols read are used to determine the next state of the finite control according to δ . The transition function also determines which symbol is written to each of the k work-tapes and whether a tape head moves one cell left (L), or one cell right (R) or remains stationary (S).

Operation ceases when M enters one of the final states q_A or q_R . In the former case M is said to *accept* \mathbf{x} , corresponding to $f=1$, in the latter M is said to *reject* \mathbf{x} , which corresponds to $f=0$.

The set of input strings accepted by M is often referred to as the *language recognised* by M .

Note that we have defined computation in terms of decision problems and not arbitrary functions. These can be catered for by including an additional tape for printing output. The modifications are straightforward and are left to the reader.

The behaviour of the machine described in Defn(1.1) is deterministic since for any given configuration of current state and $k+1$ symbols scanned by the tape heads there is exactly one move that can be made using δ . Another important model, in complexity studies, is the *non-deterministic* Turing machine. The definition of this is identical to Defn(1.1) except that δ is now a function from

$$Q \times \{\Gamma \cup B\}^{k+1} \rightarrow \text{subsets of } Q \times \{\Gamma\}^k \times \{L,R,S\}^{k+1}$$

The interpretation of this being that for any single configuration there may be a choice of many moves available. An input is accepted if some sequence of moves terminates in the accept state.

These models provide the basis for more rigorously defining "difficulty", namely the *complexity measures* (Non)-deterministic Time and (Non)-deterministic Space.

Definition 1.2: Let $f: \{0, 1\}^* \rightarrow \{0, 1\}$ be a decision problem and T, S functions from \mathbf{N} to \mathbf{N} . f is computable in deterministic time T (space $S(n)$) if f can be computed by a deterministic TM which halts after at most T moves (scans at most $S(n)$ cells on any work-tape) for all inputs of size n .

f is computable in non-deterministic time T (space $S(n)$) if there is a non-deterministic TM, M , such that for any input \mathbf{x} of size n with $f=1$, \mathbf{x} is accepted by some sequence of at most T moves (\mathbf{x} is accepted by some sequence of moves which scan at most $S(n)$ cells on any work-tape). •

Since the input tape is not considered in measuring space one can sensibly consider computations which use space $S(n) < n$.

For functions T, S as above, the decision problems computable in deterministic time T (space $S(n)$) comprise the *complexity classes* $DTIME(T(n))$ (resp. $DSPACE(S(n))$). Analogously one has the classes $NTIME(T(n))$ and $NSPACE(S(n))$ for non-deterministic computation.

Abstract complexity theory is largely concerned with the properties of and relations between complexity classes, e.g hierarchy theorems dealing with which classes are properly contained in others. We conclude this brief review by describing two of the important concerns in this area.

Let:

$$P = \bigcup_{k=1}^{\infty} DTIME(n^k) ; NP = \bigcup_{k=1}^{\infty} NTIME(n^k)$$

P corresponds to the class of deterministic polynomial time computable decision problems, which are usually regarded as being the only problems with feasible algorithms. The class NP contains

many classical combinatorial and optimisation problems for which no efficient deterministic solution is known; it can be regarded as the class of polynomial time *verifiable* problems. For example the problem of deciding whether a given n -vertex graph is 3-colourable is in NP , since one may non-deterministically "guess" a 3-colouring and (deterministically) check if it is correct. The "guessing" stage avoids the combinatorial explosion involved in checking every individual colouring in turn. No deterministic method of achieving this is known.

It is clear that $P \subseteq NP$, and although it seems probable that $P \neq NP$ this has yet to be proved. This could be shown by exhibiting a decision problem in the class $NP - P$. The seminal paper of Cook (1971) introduced the concept of NP -complete decision problems, which are essentially the most difficult problems in NP . A decision problem f is NP -complete if:

NP1)

f is in NP .

NP2)

$\forall g$ in NP there exists a deterministic polynomial time computable function, τ , which transforms instances \mathbf{x} of g to polynomially larger instances of f in such a way that g holds if and only if f holds.

If such a transformation exists then g is said to be *polynomially reducible to f* ($g \leq_p f$). This relation is transitive, so to prove a decision problem NP -complete it is sufficient to exhibit a transformation from a known NP -complete language. With these concepts $P \neq NP$ if and only if some NP -complete f is not in P .

Another important containment issue concerns the relation between space efficient computation and P . Define

$$DLOGSPACE = DSPACE(\lceil \log n + 1 \rceil)$$

Note: All logarithms are to the base 2, unless otherwise stated.

Just as P corresponds to the set of decision problems for which fast sequential algorithms exist, so $DLOGSPACE$ seems to embody the class of decision problems for which efficient *parallel time* algorithms exist. It is known that $DLOGSPACE \subseteq P$ and again the inclusion is thought to be proper. Cook (1974) introduced the class of problems $LOGSPACE$ -complete for P via logspace transformation. This is defined similarly to the property NP -complete except that the transformation must be computable in $DLOGSPACE$. The containment issue of P versus $LOGSPACE$ can be formulated as $P \neq DLOGSPACE$ if and only if some $LOGSPACE$ -complete f is not in $DLOGSPACE$.

The class $PSPACE$ consist of those decision problems computable in polynomial space. By the result of Savitch (1970), it is not necessary to distinguish between deterministic and non-deterministic machines for this class. It can be shown that $DLOGSPACE \neq PSPACE$ and so at least one of the inclusions below is strict.

$$DLOGSPACE \subseteq P \subseteq NP \subseteq PSPACE$$

The model of computation we are concerned with in this text is Boolean networks which will be formally introduced in Section(1.3). The relations between TMs and this model will be considered in detail in the next chapter.

1.2) Boolean Functions and Boolean Algebra

\mathbf{X}_n denotes an n -tuple of Boolean variables $\langle x_1, x_2, \dots, x_n \rangle$, i.e. variables taking values from $\{0,1\}$ (equivalently $\{\text{False}, \text{True}\}$). Any function $f(\mathbf{X}_n): \{0,1\}^n \rightarrow \{0,1\}^m$ is called an n -input, m -output Boolean function over \mathbf{X}_n . For brevity the cases where $m=1$ will simply be referred to as n -input Boolean functions or just Boolean functions where there is no risk of ambiguity. These correspond to the decision problems of the previous section, restricted to inputs of size n . $B_{n,m}$ will denote the set of all n -input, m -output Boolean functions and B_n the set of n -input Boolean functions. It is easy to show that $|B_{n,m}| = 2^{m2^n}$ and hence $|B_n| = 2^{2^n}$. Table(1.1) gives the 16 functions in B_2 defined in terms of arithmetic over the 2 element field $\mathbf{GF}(2)$.

A quite frequently used representation of Boolean functions is a *truth-table*, in which the value of the function for each possible input assignment is given explicitly. The examples in Table(1.2) are the truth tables for the functions \wedge , \oplus in B_2 and \neg (negation) in B_1 . In general a truth table for f in $B_{n,m}$ has 2^n rows, one for every input assignment, and $n+m$ columns, n for the values of \mathbf{X}_n and m for output values.

x_1	x_2	\wedge	\oplus	\bar{x}_1
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	0	0

Table(1.2)

Symbol	Name	f
0	Constant	0
1	Constant	1
π_1	Projection	x_1
π_2	Projection	x_2
$\bar{\pi}_1$	Projection	$1 + x_1$
$\bar{\pi}_2$	Projection	$1 + x_2$
\wedge	Conjunction (AND)	$x_1 \cdot x_2$
\vee	Disjunction (OR)	$x_1 + x_2 + x_1 \cdot x_2$
$\neg\wedge$	NAND	$1 + x_1 \cdot x_2$
$\neg\vee$	NOR	$(1 + x_1) \cdot (1 + x_2)$
\Rightarrow	Implication	$1 + x_1 + x_1 \cdot x_2$
\Leftarrow	Implication	$1 + x_2 + x_1 \cdot x_2$
$\Rightarrow\Leftarrow$	Implication	$x_1 + x_1 \cdot x_2$
$\Leftarrow\Rightarrow$	Implication	$x_2 + x_1 \cdot x_2$
\Leftrightarrow	Equivalence	$1 + x_1 + x_2$
\oplus	Exclusive-or	$x_1 + x_2$

The 16 Functions In B_2

Table (1.1)

Given any $\alpha = \langle a_1, \dots, a_n \rangle$ in $\{0, 1\}^n$, f is the result obtained by fixing $x_i = a_i$ for each $1 \leq i \leq n$. α is called an *assignment* to \mathbf{X}_n . An assignment α is said to *satisfy* f if $f = 1$.

A *partial assignment*, π is an assignment of constants to some subset of \mathbf{X}_n . $|\pi|$ denotes the number of variables fixed by π and f^π the function in $B_{n-|\pi|}$ obtained as a result. f^π is called a *subfunction*

of f . $f \in B_n$ is said to be *non-degenerate* if for all $x \in \mathbf{X}_n$, $f^{[x:=0]} \neq f^{[x:=1]}$ and *degenerate* otherwise. \hat{B}_n denotes the set of non-degenerate Boolean functions. It is easy to show that $|B_n - \hat{B}_n| = o(|B_n|)$.

Order relations $\leq, <$ are defined over B_n as follows:

$$f \leq g \iff \forall \alpha \in \{0, 1\}^n \quad f = 1 \implies g = 1$$

$$f < g \iff f \leq g \text{ and } \exists \alpha \in \{0, 1\}^n$$

for which $f = 0$ but $g = 1$

It is easy to verify that \leq is a partial order i.e $f \leq f$; $f \leq g$, $g \leq h \implies f \leq h$; $f \leq g$, $g \leq f \implies f = g$.

(P1)-(P9) below describe some of the fundamental properties of the Boolean operations \wedge, \vee and \neg . f, g , and h are arbitrary Boolean functions.

- P1) i) $f \wedge g = g \wedge f$; (*Commutativity*)
 ii) $f \vee g = g \vee f$
- P2) i) $(f \wedge g) \wedge h = f \wedge (g \wedge h)$; (*Associativity*)
 ii) $(f \vee g) \vee h = f \vee (g \vee h)$
- P3) i) $f \wedge (g \vee h) = (f \wedge g) \vee (f \wedge h)$; (*Distributivity*)
 ii) $f \vee (g \wedge h) = (f \vee g) \wedge (f \vee h)$
- P4) i) $f \wedge \bar{f} = 0$; (*Complement*)
 ii) $f \vee \bar{f} = 1$
- P5) i) $f \wedge f = f$; (*Idempotency*)
 ii) $f \vee f = f$
- P6) i) $f \wedge 0 = 0$; (*Constant*)
 ii) $f \vee 1 = 1$

- P7) i) $f \vee 0 = f$; (*Identity*)
 ii) $f \wedge 1 = f$
- P8) i) $\overline{(f \wedge g)} = \bar{f} \vee \bar{g}$; (*De Morgan's Laws*)
 ii) $\overline{(f \vee g)} = \bar{f} \wedge \bar{g}$
- P9) i) $f \wedge (f \vee g) = f$; (*Absorption*)
 ii) $f \vee (f \wedge g) = f$

The correctness of (P1-P9) is easily established by inspecting the truth-table for each relation. An easy induction on the number of operations shows that De Morgan's Laws can be generalised to arbitrarily long finite expressions.

Observe that the properties are divided into pairs in which the second is obtained by interchanging \wedge and \vee , 0 and 1 in the first. This is the so-called *principle of duality* and is a frequently applied technique in deriving identities for Boolean functions. Property(P9) may be more generally stated as:

If $f \leq g$ then: $f \wedge g = f$ and $f \vee g = g$. This will be the form used subsequently.

The *dual* of a Boolean function f is the Boolean function:
 $\tilde{f}(\mathbf{X}_n) = \bar{f}(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$

A function is *self-dual* if $\tilde{f} = f$.

f is *affine* if f may be expressed in the form:

$$a_0 \oplus \bigoplus_{i=1}^n y_i$$

where a_0 is a constant and y_i is either a constant or the variable x_i .

We now consider more concise methods of specifying Boolean functions. A *literal* is either a variable x_i or its complement \bar{x}_i . Let $\Omega = \{w_1, w_2, \dots, w_r\} \subseteq B_k$. A Boolean Ω -expression over \mathbf{X}_n is any

expression constructed according to the rules below:

- C1) $\forall x_i \in \mathbf{X}_n$ x_i is an Ω -expression.
 C2) If $w \in \Omega$ and e_1, e_2, \dots, e_k are Ω -expressions then $w(e_1, \dots, e_k)$ is also an Ω -expression.

Clearly any Ω -expression represents a unique n -input Boolean function.

Ω is called a *logical basis*. A basis is said to *cover* $F_n \subseteq B_n$ if every f in F_n is represented by some Ω -expression. A basis is *complete* if it covers B_n for every n . A distinction is made between complete bases which contain constant functions (weak-complete) and those which do not (strong-complete). The problem of characterising complete bases was solved by Post (1941). The result is stated below without proof:

Fact 1.1: Ω is a complete basis if and only if Ω satisfies all of:

- i) $\exists w$ in Ω such that $w(0, 0, \dots, 0) = 1$.
- ii) $\exists w$ in Ω such that $w^{|x_i=0} \not\leq w^{|x_i=1}$ for some x_i $1 \leq i \leq k$.
- iii) $\exists w$ in Ω such that w is non-affine.
- iv) $\exists w$ in Ω such that w is not self-dual.
- v) $\exists w$ in Ω such that $w(1, 1, \dots, 1) = 0$. \square

Two other methods of representing Boolean functions are *Disjunctive Normal Form* (DNF) and *Conjunctive Normal Form* (CNF).

A *product* is a function of the form $y_1 \wedge y_2 \wedge \dots \wedge y_k$, where $\{y_1, \dots, y_k\}$ is a set of literals not containing both x_i and \bar{x}_i for any i .

A *sum* is a function of the form $y_1 \vee \dots \vee y_k$. Conventionally a product (sum) containing no literals is defined to be the constant function 1 (0).

The associativity of \wedge and \vee ensures that these expressions are unambiguous. Henceforward we shall omit the explicit use of \wedge in specifying products and regard \wedge as having greater precedence than \vee to avoid excessive use of brackets.

Now consider any Boolean function f . Suppose $\{\alpha^{(1)}, \dots, \alpha^{(r)}\}$ is the (finite) set of assignments which satisfy f . Clearly:

$$f = p_1 \vee p_2 \vee \dots \vee p_r$$

where $p_i = 1$ if and only if the assignment to \mathbf{X}_n is $\alpha^{(i)}$. Each p_i can be represented as a product of n literals, y_1, \dots, y_n where y_j is x_j if $a_j^{(i)}$ is 1 and \bar{x}_j otherwise.

Disjunctive Normal Form is the representation of f as a sum of products. The method above shows that any Boolean function may be so expressed. CNF involves writing f as a product of sums. By considering those assignments which make $f = 0$, f may always be expressed in this way.

There are in general many different ways of expressing f in DNF or CNF. Several methods of minimising the number of product terms were developed in the field of switching theory. These are centered around the concept of prime implicants and prime clauses.

A product of literals, p , is an *implicant* of f if $p \leq f$. p is a *prime implicant* if $p \leq f$ and no product of any proper subset of the literals defining p is an implicant of f . Similarly any sum of literals, q , for which $f \leq q$ is an *implicand* of f , minimal implicands being called *prime clauses* of f .

For any $f \in B_n$, $\mathbf{PI}(f)$ will denote the set of prime implicants of f and $\mathbf{PC}(f)$ the prime clauses.

Fact 1.2: $\forall f \in B_n$

$$i) f = \bigvee_{p \in \mathbf{PI}(f)} p$$

$$ii) f = \bigwedge_{q \in \mathbf{PC}(f)} q$$

Proof: i) From the definition of implicant it is clear that the right-hand side is $\leq f$. To show $f \leq \bigvee_{p \in \mathbf{PI}(f)} p$, let $\alpha \in \{0, 1\}^m$ such that $f = 1$ and consider the product, m , of literals $\{y_1, \dots, y_n\}$ which is 1 if and only if the assignment to \mathbf{X}_n corresponds to α . Certainly m is an implicant of f . The definition of prime implicant establishes that there is some subset of the literals defining m whose product m' is in $\mathbf{PI}(f)$. Since $m \leq m'$ this proves (i).

ii) Follows by a dual argument since

$$\mathbf{PI}(\tilde{f}) = \bigcup \{ \tilde{q} : q \in \mathbf{PC}(f) \} \quad \square$$

An important problem in switching theory is to construct a minimal (number of products) DNF representation of f from its truth-table. Karnaugh maps (Karnaugh, 1953) and the tabular method of Quine (1952), (1955) and McCluskey (1956) are techniques which are feasible only for small values of n . The corresponding decision problem (i.e Does f have a DNF representation which contains at most K products, for some specified K ?) is known to be *NP*-complete as shown by Gimpel (1965) and Masek (1978).

Another normal form, first proposed by Zhegalkin (1927) is the *ringsum expansion*. This uses the weak-complete basis $\{1, \wedge, \oplus\}$ in B_2 as opposed to the strong-complete basis $\{\wedge, \vee, \neg\}$ employed in the normal forms above. The ringsum expansion can be constructed for any f as follows.

Let $\{\alpha^{(1)}, \dots, \alpha^{(r)}\}$ be the set of assignments which satisfy f and p_1, \dots, p_r the products of literals defined from these as previously. Then:

$$f = p_1 \oplus p_2 \oplus \dots \oplus p_r \quad (1.1)$$

since for any assignment to \mathbf{X}_n at most one of these products is true.

Occurrences of negated variables can be eliminated by using the identity $\bar{x} = x \oplus 1$. Finally using the fact that \wedge distributes over \oplus , (1.1) reduces to:

$$f = a_0 \oplus m_1 \oplus m_2 \oplus \dots \oplus m_t$$

where a_0 is a constant and each m_i is a product of un-negated variables. Such a product will subsequently be referred to as a *monom* and the dual construct for sums as a *clause*. For a monom m , *var* will denote the set of variables in \mathbf{X}_n which occur in m ; similarly *var* will denote the set of variables in a clause c .

An important subset of B_n is the class of *monotone* Boolean functions, M_n . f is monotone if and only if:

$$\forall x_i \in \mathbf{X}_n \quad f^{|x_i=0} \leq f^{|x_i=1}$$

Lemma 1.1: If $f \in M_n$ then no prime implicant of f contains a negated variable.

Proof: Let $f \in M_n$ and without loss of generality suppose $m = \bar{x}_1 \wedge p(\mathbf{X}_n - \{x_1\})$ is a prime implicant of f . Since $\bar{x}_1 \wedge p \leq f$ and f is monotone so $x_1 \wedge p \leq f$ (by considering any assignment which fixes all of the literals in p to 1). It follows that $\bar{x}_1 p \vee x_1 p \leq f$ but

$$\bar{x}_1 p \vee x_1 p = (\bar{x}_1 \vee x_1) \wedge p = 1 \wedge p = p$$

and this contradicts the assumption that m was a prime implicant. \square

A similar result holds for prime clauses of monotone Boolean functions.

Corollary 1.1:

- i) $\forall f \in M_n \quad \bigvee_{p \in \mathbf{PI}(f)} p$ is the minimal DNF representation of f .
- ii) $\forall f \in M_n \quad \bigwedge_{q \in \mathbf{PC}(f)} q$ is the minimal CNF representation of f .

Proof:

i) Suppose $h = \bigvee_{i=1}^r m_i$ is a DNF representation of f which contains fewer than $|\mathbf{PI}(f)|$ products. From Lemma(1.1) it may be assumed that no product of h contains a negated variable. Since $r < |\mathbf{PI}(f)|$ there must be some prime implicant of f , p say, which is not in $\{m_1, \dots, m_r\}$. Consider the assignment α to \mathbf{X}_n which sets exactly var to 1 and $\mathbf{X}_n - \text{var}$ to 0. By the assumption that $h = f$, $h = 1$ and so some m in $\{m_1, \dots, m_r\}$ must equal 1 under α , but then $\text{var} \subset \text{var}$ and now by changing α so that additionally the variables in $\text{var} - \text{var}(m)$ are all 0 a contradiction results since $h(\alpha) = 1$ but $f(\alpha) = 0$.

ii) Duality. \square

Lemma 1.2: Let $\Omega = \{\wedge, \vee, 0, 1\} \subseteq B_2$. Any Ω -expression $e(\mathbf{X}_n)$ specifies a monotone Boolean function.

Proof: Since 0, 1, x_i are all monotone it is sufficient to prove that $f \vee g$ and $f \wedge g$ are both monotone if f and g are monotone. Only the former need be shown, the latter case following by a dual argument.

Let $h = f \vee g$ for any monotone f and g . Consider any x_i in \mathbf{X}_n .

$$h^{x_i=0} = f^{x_i=0} \vee g^{x_i=0}$$

From the definition of \vee , any assignment which satisfies $h^{x_i=0}$ must satisfy at least one of $f^{x_i=0}$, $g^{x_i=0}$. As f and g are both

monotone, the same assignment must satisfy $f^{x_i=1} \vee g^{x_i=1}$, and so:

$$h^{x_i=0} \leq f^{x_i=1} \vee g^{x_i=1} = h^{x_i=1} \quad \square$$

Corollary 1.2: $f \in M_n \iff \tilde{f} \in M_n$

Proof: Easily derived from Lemma(1.1), DeMorgan's Laws and Lemma(1.2) \square

Combining Lemma(1.1) and Lemma(1.2) yields:

Theorem 1.1: f is monotone if and only if f can be represented by a $\{\wedge, \vee, 0, 1\}$ -expression. \square

Another important subset of B_n is the class of *symmetric* Boolean functions. f is symmetric if its output only depends on the number of inputs which are true. Thus for any permutation σ of $\langle 1, 2, \dots, n \rangle$

$$f = f \circ \sigma$$

if f is symmetric.

S_n will denote the class of n -input symmetric Boolean functions. Any f in S_n can be succinctly described by a binary word of length $n+1$, $\mathbf{w} = w_0 w_1 \cdots w_n$, the w_i bit giving the value of f when exactly i inputs are 1. \mathbf{w} is called the *spectrum* of f . By considering the number of distinct spectra it follows that there are 2^{n+1} functions in S_n . Examples of symmetric Boolean functions are:

$$\mathbf{C}_k^n(\mathbf{X}_n) = 1 \iff \sum_{i=1}^n x_i = 0 \pmod{k}$$

$$\mathbf{E}_k^n(\mathbf{X}_n) = 1 \iff \sum_{i=1}^n x_i = k$$

$$T_k^n(\mathbf{X}_n) = 1 \iff \sum_{i=1}^n x_i \geq k$$

In all cases the summation is arithmetic. T_k^n is the k -th *threshold function*. These are the class of monotone symmetric functions. $T_{n/2}^n$ is the *majority function* (denoted MAJ_n).

Arithmetic functions can easily be represented as multiple output Boolean functions by encoding the input data in binary e.g if \mathbf{X}_n and \mathbf{Y}_n are disjoint n -tuples of Boolean variables then:

$$ADD(\mathbf{X}_n, \mathbf{Y}_n) \{0, 1\}^{2n} \rightarrow \{0, 1\}^{n+1}$$

denotes the Boolean representation of integer addition.

Graph-theoretic problems are normally encoded using an adjacency matrix to represent an n -vertex graph. Thus let

$$\mathbf{X}_n^U = \{x_{ij} : 1 \leq i < j \leq n\}; \mathbf{X}_n^D = \{x_{ij} : 1 \leq i, j \leq n\}$$

\mathbf{G}_U is a function from \mathbf{X}_n^U to n -vertex undirected graphs, $\mathbf{G}_U(\mathbf{X}_n^U)$ contains an edge $\{i, j\}$ if and only if x_{ij} in \mathbf{X}_n^U is 1. $\mathbf{G}_D(\mathbf{X}_n^D)$ is the n -vertex directed graph defined in a similar manner.

1.3) Boolean Networks

Let $\Omega \subseteq B_2$. A *Boolean Ω -network*, T , is a directed acyclic graph containing 2 disjoint sets of nodes; I is the set of nodes with in-degree 0 (the *inputs* of T); G is the set of nodes with in-degree 2 (the *gates* of T). Each x_j in \mathbf{X}_n is associated with exactly one input node i_j in I , any remaining input nodes are associated with constant functions. Each gate g of T is associated with some function h in Ω , denoted by $op(g) = h$ or g is an h -gate. If g is a gate then the inputs of g are g_L (Left) and g_R (Right). For any node v of T , the number of edges (or *wires*) leaving (entering) v is termed the fanout (fanin) of

v , and is denoted by $\phi(v)$. The fanout (fanin) of a network is the maximal fanout (fanin) of any node. Any node with fanout 0 is called an *output* of T . Ω is the *basis* of T .

If v, w are nodes such that there is a wire $\langle v, w \rangle$ from v to w , v is said to be a *predecessor* or *input* of w ; similarly w is said to be a *successor* or *output* of v . This will also be referred to as " v enters w ". A node v is an *ancestor* (resp. *descendant*) of a node w , if there is a directed path from v to w (resp. from w to v).

For brevity we shall subsequently refer to "the input x_j of T " rather than "the input of T associated with x_j " and to \mathbf{X}_n instead of I , as the inputs of T . Similarly we shall not make the ordering of gate inputs explicit, unless significant.

If $\Omega = B_2$ then T is a *combinational network*. If $\Omega = \{\wedge, \vee\}$ then T is a *monotone Boolean network*.

With each node, v , of an Ω -network T a Boolean function $res(v)(\mathbf{X}_n)$ is associated as follows:

$$res(v)(\mathbf{X}_n) = \begin{cases} 0 & \text{if } v \text{ is an input node labelled } 0 \\ 1 & \text{if } v \text{ is an input node labelled } 1 \\ x_i & \text{if } v \text{ is the input } x_i \text{ of } T \\ res(v_1) \text{ op}(v) res(v_2) & \text{otherwise} \end{cases}$$

where v_1, v_2 are the inputs of v .

An Ω -network T computes or realises $\{f_i\}$ in $B_{n,m}$ if and only if there are m nodes $\langle v_1, \dots, v_m \rangle$ in T such that $res(v_i) = f_i$. It is clear from Theorem(1.1) that f is monotone if and only if f can be computed by a monotone Boolean network.

It will be assumed that any Ω -network T computing a single output Boolean function contains exactly 1 output node t this being the unique node whose result is f .

For notational convenience the remainder of this section will be couched in terms of single output Boolean functions, however it is trivially verified that the results and concepts introduced generalise to m -output functions.

The two complexity measures of interest are *network size* and *depth*. For an Ω -network T these are respectively:

$$\mathbf{C}_\Omega(T) = |\{ g : g \text{ is a gate node in } T \}|$$

$$\mathbf{D}_\Omega(T) = \text{Length of longest directed path in } T$$

It will sometimes be convenient to consider the nodes of a network as partitioned into *levels* L_0, \dots, L_D ; L_0 being the input nodes, L_i those gates which receive an input from a node in L_{i-1} and a node in L_j for some $j \leq i - 1$.

The nodes of a network are said to be labelled in *topological order* if each node, u , is assigned a distinct number n such that: for all edges $\langle u, v \rangle$ in the network the numbering satisfies $n < n(v)$.

For any Boolean function f :

$$\mathbf{C}_\Omega(f) = \min \{ \mathbf{C}_\Omega(T) : T \text{ computes } f \}$$

$$\mathbf{D}_\Omega(f) = \min \{ \mathbf{D}_\Omega(T) : T \text{ computes } f \}$$

where $\Omega = B_2$ these will be denoted simply by $\mathbf{C}(f)$ and $\mathbf{D}(f)$. $\mathbf{C}(f)$ is the *combinational complexity* of f .

An Ω -network T , computing f , is optimal if $\mathbf{C}_\Omega(T) = \mathbf{C}_\Omega(f)$. The following lemma summarises some important properties of optimal combinational networks.

Lemma 1.3: Let T be an optimal combinational network computing $f \in B_n$, where T contains at least one gate node. T satisfies all of the

following:

i) No two nodes of T compute the same function of \mathbf{X}_n

ii) Every gate computes a function of the form:

$$(y_1^a \wedge y_2^b)^c \text{ (}\wedge\text{-type) or } (y_1 \oplus y_2)^d \text{ (}\oplus\text{-type)}$$

where $\{a, b, c, d\} \in \{0, 1\}$ and $y^e = y$ if $e = 1$ and \bar{y} otherwise.

iii) No gate receives two inputs from the same node.

Proof:

i) If v_1 is a node and v_2 is a gate such that $res(v_1)(\mathbf{X}_n) = res(v_2)(\mathbf{X}_n)$ then the network T' constructed by deleting v_2 together with all wires $\langle v_2, w \rangle$ from T and adding wires $\langle v_1, w \rangle$ still computes f but contains one fewer gate.

ii) The only functions in B_2 which are neither \wedge -type nor \oplus -type are constant functions or projections. It will be shown that nodes computing constant functions or gates computing projections may be eliminated by "absorbing" their result into successor gates.

Suppose $res(u) \in \{0, 1\}$ for some node u of T . Let v be any successor of u and w the other input of v . Then: $res(v) \in \{0, 1, res(w), \overline{res(w)}\}$. In the first two cases the gate v may be deleted, replaced by a constant function and the wires $\langle u, v \rangle$, $\langle w, v \rangle$ removed from T . In the remaining cases v may be replaced by the appropriate projection of (w, w) and again the wire $\langle u, v \rangle$ may be deleted.

To complete the proof of (ii) it remains to eliminate instances of projections. Clearly any projection of the form π_1, π_2 can be removed, so it is sufficient to consider only projections of the form $\bar{\pi}_1$. Let $op(u) = \bar{\pi}_1$; r, s be the inputs of u ; and v and w as before. In this case:

$$\begin{aligned} \text{res}(v) &= \overline{\text{res}(r)} \text{op}(v) \text{res}(w) \\ &= b(\text{res}(r), \text{res}(w)) \end{aligned}$$

for some $b \in B_2$ depending on $\text{op}(v)$.

Now T can be re-wired as follows: Delete the wire $\langle u, v \rangle$; add a wire $\langle r, v \rangle$; replace v by a b -gate.

In both cases every wire $\langle u, v \rangle$ is eventually removed so u has fan-out 0 and can be eliminated.

iii) $\forall h \in B_2 \ h \in \{0, 1, g, \bar{g}\}$. Thus any gate both of whose inputs are from the same node can be replaced by a constant function or projection and thence from (ii) eliminated. \square

The definition above restricts consideration to networks with fan-in 2, but allows unlimited fanout. The limitation on fan-in is easily justified since in practical terms it is costly to manufacture gates with large numbers of inputs. Furthermore since a finite number of 2-input gates can compute any function in B_k , permitting larger constant fanin could only reduce combinational complexity and depth by a constant factor.

The next two results indicate that these complexity measures are fairly insensitive to the choice of complete basis used and to restricting node fanout to be at most 2.

Lemma 1.4: Let Ω_1 and Ω_2 be complete logical bases from B_2 and let f be a function in B_n . There are constants s and d such that:

$$\mathbf{C}_{\Omega_1}(f) \leq s \mathbf{C}_{\Omega_2}(f)$$

$$\mathbf{D}_{\Omega_1}(f) \leq d \mathbf{D}_{\Omega_2}(f)$$

Proof: Let T be an optimal Ω_2 -network realising f and let

$$REM = \Omega_2 - \Omega_1 \subset B_2$$

Since Ω_1 is logically complete, any function b in REM can be computed by some 2-input Ω_1 -network, which contains at most s gates and has depth at most d , for some constants s and d depending on Ω_1, Ω_2 . It follows that T can be transformed into an Ω_1 -network by replacing each gate u , with $op(u)$ in REM , by the appropriate Ω_1 -network. Clearly this increases the size and depth of T by at most the factors stated. \square

Lemma 1.5: Let T be an Ω -network computing $f \in B_{n,m}$. For any constant $t \geq 2$ there is an $\{\Omega \cup I\}$ -network T' , where I denotes the single argument identity function, which satisfies:

- i) T' computes f .
- ii) $\mathbf{C}_{\{\Omega \cup I\}}(T') \leq \left(1 + \frac{1}{(t-1)}\right) \mathbf{C}_{\Omega}(T) + \frac{(m-1)}{(t-1)}$
- iii) $\mathbf{D}_{\{\Omega \cup I\}}(T') \leq (1 + \log_t 2) \mathbf{D}_{\Omega}(T) + \log_t m$
- iv) Every node, u , of T' has fanout at most t .

Proof: See Hoover, Klawe, Pippenger (1984). \square

One important class of restricted fanout models are Boolean formulae.

A *formula* over the basis Ω is defined in the same way as a Boolean Ω -network except that gate nodes have fanout at most one.

The set of Ω -formulae is isomorphic to the set of Ω -expressions defined previously. Noting this correspondence it will frequently be more convenient to adopt the following inductive definition of Ω -formula.

Definition 1.3: Let $\Omega \subseteq B_2$. An Ω -formula L over \mathbf{X}_n is any expression generated by repeated application of the the rules below:

- i) $\forall x_i \in \mathbf{X}_n$ x_i is an Ω -formula.
- ii) $c \in \{0, 1\} \cap \Omega$ is an Ω -formulae.
- iii) If $\{\bar{\pi}_1, \bar{\pi}_2\} \cap \Omega \neq \{\}$ and L is an Ω -formula, then $\neg L$ is also an Ω -formula.
- iv) If L is an Ω -formula then (L) is also an Ω -formula.
- v) If $*$ $\in \Omega$ which depends on both its arguments (i.e is not a constant function or projection) and L_1, L_2 are Ω -formulae then $L_1 * L_2$ is an Ω -formula. •

$\mathbf{L}_\Omega(G)$ will denote the number of 2-input gates occurring in an Ω -formula G , this being precisely one less than the total fanout from the input nodes. In terms of Defn(1.3), $\mathbf{L}_\Omega(G)$ is the number of times rule (v) is applied in constructing G . $\mathbf{L}_\Omega(f)$ will denote the Ω -formula size of a Boolean function f . As before, if $\Omega = B_2$, then the notation $\mathbf{L}(G)$, $\mathbf{L}(f)$ will be used. Formulae are examined extensively in Chapter(3).

Finally we consider a significant difference between TM computation and Boolean networks.

Let $f: \{0, 1\}^* \rightarrow \{0, 1\}$ be a decision problem. The *family* of Boolean functions corresponding to f is the infinite sequence

$$[f_n] = \langle f^{(1)}, f^{(2)}, \dots, f^{(n)}, \dots \rangle$$

$f^{(n)}$ is the Boolean function obtained by restricting f to inputs of size n . A family of Boolean functions has Ω -network size (or depth) $G(n)$ iff

$$\forall n \quad \mathbf{C}_\Omega(f^{(n)}) = G(n) \quad (\text{or } \mathbf{D}_\Omega(f^{(n)}) = G(n))$$

We define the complexity classes $\Omega - SIZE(G(n))$, resp. $\Omega - DEPTH(G(n))$ as being the sets of families of Boolean functions computable with Ω -network size (depth) at most $G(n)$. Thus a family of Boolean functions is regarded as being computed by a sequence $\langle T_1, T_2, \dots, T_n, \dots \rangle$ of Boolean Ω -networks, T_n realising $f^{(n)}$, the network complexity of f being defined with respect to this sequence. However the TM complexity of f is defined with respect to a *single* TM, viz $f \in DTIME(T(n))$ if there exists a DTM M which accepts f and makes at most T moves on any input of length n .

Formally this situation is described by saying that TMs are a *uniform* model and Boolean networks a *non-uniform* model of computation.

All decision problems may be solved by networks but it is well known that some decision problems are not *TM*-computable. A less extreme consequence of non-uniform behaviour is that there are decision problems which may be solved much more efficiently by networks than by TMs and so reasonable simulations of networks by TMs cannot exist, see e.g Meyer and Stockmeyer (1973), Fischer and Rabin (1974). Uniform circuit complexity theory attempts to rectify this situation by constraining the members of a family of networks to be "similar". e.g a family of networks computing $[f_n]$ is uniform if there is a DTM which given n , encoded in unary, can construct some standard encoding of the n 'th network within some time bound depending on the network size. This is one of the most rapidly expanding areas of complexity theory and an adequate description is beyond the scope of this text. The interested reader is referred to the paper of Borodin(1975) which introduces some of the fundamental concepts and those of Cook (1979), Pippenger (1980) and Ruzzo (1981).

Bibliographic Notes

For further background on computational complexity theory the reader is referred to Aho, Hopcroft and Ullman (1974) and Hopcroft and Ullman (1979). Machtey and Young (1978) is an advanced level description of modern abstract complexity theory. Garey and Johnson (1979) presents a comprehensive account of *NP*-completeness. Classical switching theory is surveyed in the texts of Friedman (1975), Harrison (1965) and Miller (1965).

There are a number of fields of relevance to Boolean network complexity which will not be examined in any detail in this book. The most important of these are algebraic complexity, which studies computational complexity of networks in which general arithmetic functions are available as base operations, this area being covered in the text of Borodin and Munro (1975); and also VLSI complexity as described in Ullman (1984).

Chapter 2

Combinational Network Complexity

..... *I shall tell you*

A pretty tale. It maybe that you have heard it;

But, since it serves my purpose, I will venture

To stale't a little more

Coriolanus I, i, 92 – 95

Cori-

Combinational networks as introduced above are the basic computational model examined in this book. The present chapter is mainly concerned with relations involving network complexity measures and also the combinational complexity of some particular Boolean functions.

In Section(2.1) simulations of Turing Machines by combinational networks are considered; the main results presented being the theorem of Fischer & Pippenger, which relates *DTIME* to network size, and that of Borodin relating *NSPACE* to network depth. Following this various relations amongst network complexity measures are considered. The section concludes with a description of the Skyum and Valiant (1984) results concerning reductions between families of Boolean functions.

Section(2.2) is concerned with estimating the worst-case complexity of Boolean functions. The important theorem of Shannon, showing that "almost all" functions in B_n have combinational complexity $\Omega(2^n/n)$ is proved here. A matching upper bound is provided using a construction due to Lupanov. In the same vein general upper and lower bounds on network depth are given. The theorem of Paterson and Valiant relating network size and depth is expounded in

Section(2.3).

The concluding sections of this chapter deal with the combinational complexity of some specific Boolean functions: The lower bound arguments of Schnorr, Paul, Stockmeyer and Blum are examined in Section(2.4); efficient networks for certain arithmetic and all symmetric functions being given in Section(2.5).

2.1) Simulation Results

The complexity measures deterministic TM-time and space reflect intuitive notions of the temporal and spatial requirements of particular computations. In this section the relation between these measures and combinational network complexity is examined. First some further terminology is required.

A DTM, M , is *oblivious* if the vector $\langle h_1, \dots, h_{k+1} \rangle$ of tape head positions depends solely on the input size and the number of moves made. As before let $f^{(n)}$ denote the restriction of some decision problem, f , to inputs of size n . A relation between deterministic TM-time and network size is established in two stages:

SIM1)

By relating $C(f^{(n)})$ to the time complexity of an oblivious DTM computing f .

SIM2)

By exhibiting an efficient simulation of arbitrary DTMs by oblivious DTMs.

Both simulations were first presented by Fischer & Pippenger (1979); Our description follows that of Schnorr (1976a) in which a careful analysis of constant factors is made.

It is convenient to view the transition function, δ , as describing a simple program consisting of numbered instructions falling into one of the following categories.

- 1) *Actions*: (s, F, t) : interpreted as instruction number s consists of performing some action F followed by a jump to instruction number t . Here F is one of:
 - a) Move the head on tape j right (R_j)
 - b) Move the head on tape j left (L_j)
 - c) The head on tape j prints $e \in \{0, 1\}$ ($P_j(e)$)
 - d) The head on tape j prints the symbol scanned by the head on tape i ($TR_j(i)$)
 - e) Halt (in which case $t = s$)
- 2) *Tests*: (s, T, r, t) : interpreted as instruction number s is "If T then go to r else go to t ". Tests, T , are of the form:

$Q_j(e)$ Does the head on tape j observe $e \in \{0, 1, B\}$?

Clearly any move of δ can be encoded as a finite sequence of Actions and Tests and so the transition function gives rise to a program which mimics the behaviour of M . For an arbitrary DTM, M , p will denote the corresponding program and $|p|$ the number of instructions contained therein. Instruction number 0 is the initial instruction, and since we are considering decision problems, two instructions s_A , s_R are identified as halt and accept, halt and reject. When discussing the execution of such a program we shall refer to the number labelling the current instruction as the *state of p* . $T_p(n)$ will denote the worst-case running time of p on inputs of size n .

Theorem 2.1: (Schnorr 1976a) Let $f: \{0, 1\}^* \rightarrow \{0, 1\}$ be any decision problem. \forall oblivious Turing programs, p , which compute f :

$$\mathbf{C}(f^{(n)}) \leq 7|p|T_p(n)$$

Proof: Since p is oblivious there is a function, $pos(i, j, n)$, $0 \leq i \leq T_p(n)$, $1 \leq j \leq k+1$ which gives the position of the j 'th tape head after i steps on all inputs of size n . Consider the following n -input Boolean functions, where $\alpha \in \{0, 1\}^n$:

$A(i, s)(\alpha) \iff$ After i steps with input α p is in state s

$B(i, l, j, e)(\alpha) \iff$ After i steps with input α e is in cell l of tape j

For each i $0 \leq i \leq T_p(n)$ define C_i to be the set of Boolean functions over \mathbf{X}_n :

$$C_i = \{A(i, s), B(i, l, j, e) : \forall s, l, j, e\}$$

The program terminates in one of the final states s_A, s_R . From the definition of $A(i, s)$ above we have:

$$f^{(n)}(\mathbf{X}_n) = A(T_p(n), s_A)(\mathbf{X}_n)$$

It is thus sufficient to prove that $\forall 0 \leq i \leq T_p(n)$

$$\mathbf{C}(C_i) \leq 7|p|i$$

Every function in C_0 is either a variable x_i or a constant function so $\mathbf{C}(C_0) = 0$. Assume it has been established that $\mathbf{C}(C_i) \leq 7|p|i$ for some $0 \leq i < T_p(n)$. It must be shown that:

$$\mathbf{C}(C_{i+1}) \leq \mathbf{C}(C_i) + 7|p|$$

i.e given a combinational network computing C_i , the functions $A(i+1, s) \forall$ states s , and $B(i+1, l, j, e) \forall l = pos(i, j, n)$ can be computed using no more than $7|p|$ 2-input gates. Obviously $B(i+1, l, j, e) = B(i, l, j, e)$ if $l \neq pos(i, j, n)$.

$A(i+1, s)$ is equal to,

$$\begin{aligned} & \bigvee_{(t, F, s) \in p} A(i, t) \vee \\ & \bigvee_{\substack{(t, Q_j(e), s, r) \in p \\ \wedge l = \text{pos}(i, j, n)}}} A(i, t) \wedge B(i, l, j, e) \vee \\ & \bigvee_{\substack{(t, Q_j(e), r, s) \in p \\ \wedge l = \text{pos}(i, j, n)}}} A(i, t) \wedge \overline{B(i, l, j, e)} \end{aligned}$$

Thus the state at time $i+1$ is s if and only if the state at time i is t and t is an action or test ending in state s .

If ac , te denote the number of actions and tests in p then it is easy to see that the set of functions $\{A(i+1, s) : \forall s \in p\}$ can be computed using at most $ac + 4te$ gates from C_i .

The functions $B(i+1, l, j, e)$ where $l = \text{pos}(i, j, n)$ can be represented by:

$$B(i+1, l, j, e) = \text{Changed} \vee B(i, l, j, e) \wedge \text{Unchanged}$$

where:

$$\text{Changed} = \bigvee_{(r, P_j(e), s) \in p} A(i, r) \vee \bigvee_{\substack{(r, TR_j(j'), s) \in p \\ \wedge l' = \text{pos}(i, j', n)}}} B(i, l', j', e) \wedge A(i, r)$$

$$\text{Unchanged} = \bigwedge_{\substack{(r, P_j(f), s) \in p \\ r, TR_j(m), s) \in p}} \overline{A(i, r)}$$

If p_r , tr denote the number of print and transfer actions in p then, observing that $B(i, l, j, 0) = \overline{B(i, l, j, 1)}$ the sets:

$$\{\text{Changed} : \forall l, j, e\} ; \{\text{Unchanged} : \forall l, j, e\}$$

can be computed from C_i using at most $pr + 2tr$, $pr + tr$ gates

respectively. Thus all the functions:

$$\{B(i+1, l, j, e) : \forall j, e, l = \text{pos}(i, j, n)\}$$

can be derived using no more than $2pr + 3tr + 2k$ new gates.

The $2k$ term entering since $B(i, l, j, e)$ must be updated for each of the k work-tapes and this requires one \wedge -gate and one \vee -gate for each tape. Combining these bounds yields:

$$\mathbf{C}(C_{i+1}) \leq ac + 4te + 2pr + 3tr + 2k + \mathbf{C}(C_i)$$

$$\leq 6|p| + 2k + \mathbf{C}(C_i)$$

$$\leq 7|p| + \mathbf{C}(C_i)$$

This last holds since we can assume that for each tape j there is at least one instruction to print on tape j and at least one instruction to read (i.e transfer or test) from tape j , otherwise tape j plays no part in the computation. Thus the input tape, which is read-only, need not be counted in the above exposition and additionally it follows that $2k \leq |p|$.

This completes the proof that $\mathbf{C}(C_i) \leq 7|p|i$. \square

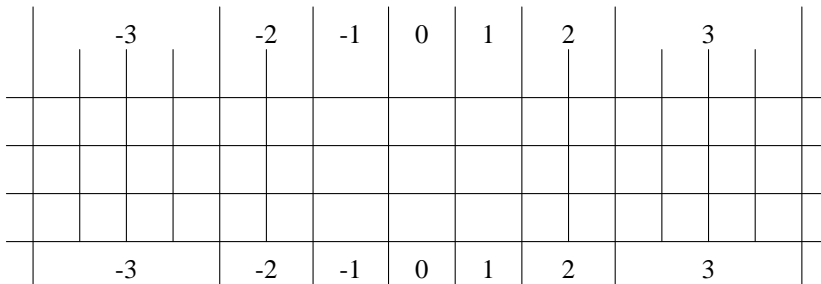
With Theorem(2.1) the combinational complexity of $[f_n]$ is seen to be only a constant factor larger than the time required by an oblivious DTM recognising the related decision problem. To obtain a relation between networks and arbitrary DTMs an efficient simulation of these by oblivious machines is used. This will entail some slight loss in speed.

Theorem 2.2: (Fischer/Pippenger 1979) For all DTMs, M , computing $f: \{0, 1\}^* \rightarrow \{0, 1\}$ \exists a DTM, OBM , such that $\forall n$:

- i) OBM is oblivious
- ii) $\forall \mathbf{x} \in \{0,1\}^n$ OBM accepts \mathbf{x} if and only if M accepts \mathbf{x} .
- iii) If p, obp are the programs arising from M and OBM respectively then:

$$T_{obp}(n) = O(T_p(n) \log T_p(n))$$

Proof: The method is essentially an oblivious version of the Henie/Stearns (1966) simulation of k -tape DTMs by 2-tape DTMs. Let M be a k -tape DTM with program p running in time $T_p(n)$. OBM is a $(k + 1)$ -tape DTM with a tape OTB_j for each tape T_j of M and one additional work-tape for temporary storage. Each tape of OBM consists of 3 tracks and is divided into segments numbered $\dots, -3, -2, -1, 0, 1, 2, 3 \dots$. Segment 0 contains 1 tape cell per track; segment i ($i \neq 0$) contains $2^{|i|-1}$ tape cells per track. The section of a track contained within a segment is termed a *block*.



The tape alphabet of OBM is $\{0, 1, B, \#\}$, $\#$ being a special "empty" symbol, the $\#$ and B symbols are printable, merely for technical convenience.

A cell within a track of OTB_j either contains a symbol in $\{0, 1, B\}$, in which case it corresponds to some cell on T_j containing the same symbol, or it contains the $\#$ symbol. A block is *full* if it

contains no # symbol, *empty* if it contains solely # symbols. A segment is *clean* if it contains 1 or 2 full blocks with the remaining blocks being empty. A segment is *saturated (voided)* if it consists of 3 full (empty) blocks.

Initially the first track of OBT_j corresponds exactly to the starting contents of T_j , all other tracks being empty. Thus all segments are clean to begin with. The simulation of the computation on T_j by OBT_j is identical for all tapes. The simulating program, *obp*, is constructed so that all of the following hold after each simulation step.

- S1) Every block is empty or full.
- S2) The content of T_j is formed by concatenating full blocks of OBT_j commencing with the smallest numbered segments and lowest track within a segment.
- S3) All tape heads scan segment 0 which contains the symbol scanned by the corresponding head in M .

The simulation process is built around a recursively defined procedure $Sim(r)$ which will satisfy (S1-S3) on completion and additionally:

- S4) $Sim(r)$ simulates 2^r steps of p , never moving outside segments $\pm(r+1)$. On completion segments $-r, -r+1, \dots, r-1, r$ are all clean. The number of full blocks within any segment changes by at most one.

$Sim(r)$ is defined as follows: $Clean(r)$ performs the actions below:

The notation "segment($\pm s$)" is a shorthand for "segment(+ s)" resp. "segment(- s)". Note that since the simulation must be oblivious the head movement must be made for both possibilities even though at most one will affect OBT_j .

```

if  $r = 0$  then
  Simulate one step of  $p$  ensuring that (S1-S4)
  hold for segments  $-1,0,1$ 
else
   $Sim(r - 1); Clean(r);$ 
   $Sim(r - 1); Clean(r)$ 
fi

if segment( $\pm r$ ) is saturated then
  combine two blocks into one block
  and copy to segment  $\pm(r + 1)$ 
else if segment( $\pm r$ ) is voided then
  bisect a block of segment  $\pm(r + 1)$ 
  and copy these blocks to segment  $\pm r$ 
fi

```

If $Sim(r)$ is correct then the simulation can be performed by activating Sim with $r = \lceil \log T_p(n) \rceil$. Correctness is proved by induction on r $0 \leq r \leq \lceil \log T_p(n) \rceil$.

First suppose that whenever $Sim(r)$ is called segments $-(r + 1), -r, \dots, r, r + 1$ are clean and that (S1-S3) hold. This is the situation that holds before any simulation is carried out, and hence on the very first call of $Sim(r - 1), \dots, Sim(0)$. Induction is used to show that (S1-S4) holds on the completion of $Sim(r)$. Clearly $Sim(0)$ can be realised so that (S1-S4) are true after $Sim(0)$. Suppose $Sim(r - 1)$ behaves correctly. To prove $Sim(r)$ works it is sufficient to show that both calls on $Clean(r)$ can be carried out. Certainly the first call can be performed since $Sim(r - 1)$ does not affect segments $\pm(r + 1)$ which are initially clean by the inductive hypothesis. The second call of $Clean(r)$ can only fail if both segments $r, r + 1$ (resp. $-r, -r - 1$) are

saturated (or voided). Prior to this second call, segments $\pm(r+1)$ are accessed only on the first call of $Clean(r)$. Suppose this call saturated (voided) segment $\pm(r+1)$. Then 2 blocks in segment $\pm r$ must be emptied (filled) by this call. By the inductive hypothesis, since (S4) holds for $Sim(r-1)$, at most one of these blocks can become full (empty) on the second call of $Sim(r-1)$. Therefore the second call of $Clean(r)$ can be carried out.

To complete the proof of the theorem it remains to describe an effective program obp which imitates $Sim(\lceil \log T_p(n) \rceil)$ (i.e one which does not use recursion) and to bound $T_{op}(n)$. Such a program can be readily described after observing that every 2^r -th simulation step is followed by a sequence of calls $Clean(1), Clean(2), \dots, Clean(r+1)$. This yields:

```

program obp
   $t := 0$ ;
  repeat
     $t := t + 1$ ;
     $Sim(0)$ ;
     $m := \max \{ i \mid 2^i \text{ divides } t \} + 1$ ;
     $Clean(r)$  for  $r$  from 1 to  $m$ 
  until finished

```

$Clean(j)$ requires at most $O(2^j)$ steps to complete and is called 2^{r-j+1} times within the first 2^r simulation steps. Thus $O(2^r \log T_p(n))$ steps are sufficient to carry out all $Clean(j) \forall 1 \leq j \leq r$.

The total number of remaining steps of obp during the first 2^r simulation steps is bounded by $O(2^r \log T_p(n))$. Thus the entire oblivious simulation can be performed in:

$$O(T_p(n) \log T_p(n)) \text{ steps} \quad \square$$

Corollary 2.1: If $f \in DTIME(T(n))$ then

$$\mathbf{C}(f^{(n)}) = O(T(n) \log T(n)) \quad \square$$

Thus lower bounds on combinational complexity of $G(n)$ say, yield lower bounds of $\Omega\left(\frac{G(n)}{\log G(n)}\right)$ on deterministic time.

We now turn to the relation between network depth and non-deterministic space. We recall that if R is a binary relation over a set A , the (reflexive) transitive closure of R is the relation:

$$R^* = I \cup R \cup R^2 \cup \dots \cup R^i \dots$$

If A is a finite set containing n elements then R can be encoded as an $n \times n$ Boolean matrix, M , in an obvious way. It is well known that R^* is encoded by the matrix M^* defined by $M^* = (I + M)^{n-1}$

Theorem 2.3: (Borodin, 1975) If f is computable by a NDTM using space, $S(n) \geq \log n$, then $f^{(n)}$ can be realised by a combinational network of depth:

$$\mathbf{D} = O(S(n)^2)$$

Proof: Let M be a NDTM computing f in space $S(n) \geq \log n$. Since it does not affect the spatial requirements it may be assumed that M has precisely one work-tape. Let x_1, \dots, x_n be the symbols on the input tape at the start of a computation, and $q = |Q|$. Since M operates within space $S(n)$ the total number of *configurations* of current state, head positions and work-tape contents is exactly $N = q \cdot n \cdot S(n) \cdot 3^{S(n)}$. Let ID denote this set of configurations and define a relation *Next* over $ID \times ID$ by:

$\langle i, j \rangle \in \text{Next} \iff$ there is a move from configuration number i to configuration number j .

$PATH$ will denote the $N \times N$ Boolean, matrix encoding $Next^*$, thus $PATH_{ij} = 1$ if and only if there exists a sequence of moves starting in configuration i and terminating in configuration j . Thus $PATH = (NEXT + I)^N$.

Given these concepts it is clear that the problem of determining whether the input $x_1 \cdots x_n$ is accepted is equivalent to determining if there is a path from the corresponding starting configuration to some accepting configuration. Consider the network of Figure(2.1) in which $\{f_1, \dots, f_r\}$ denote the accepting configurations.

It remains to specify how the inputs x_i are connected to the inputs of the transitive closure network and to establish the depth bound.

Let i be the number of any configuration in which the input tape head scans x_k .

x_k is connected to $NEXT_{ij} \iff$ there is a move from i to j only if $x_k = 1$

\bar{x}_k is connected to $NEXT_{ij} \iff$ there is a move from i to j only if $x_k = 0$

$NEXT_{ij}$ is set to 1 (0) \iff there is (is not) a move from i to j regardless of the value of x_k .

Clearly with these settings the network above computes $f^{(n)}(\mathbf{X}_n)$. Its depth is $O(S(n)^2)$ since the $PATH$ matrix can be computed using $\log N + 1$ levels of Boolean matrix product and a single product can be realised by a network of depth $\lceil \log N \rceil$ (e.g using the "obvious" Boolean matrix product network). Since the number of accepting configurations is certainly no more than N , so $\log N$ depth suffices to compute the final \vee -stage. The depth bound now follows since:

Figure 2.1

$$\log N = S(n) \log 3 + \log(q \cdot n \cdot S(n)) = O(S(n))$$

□

In combination Corollary(2.1) and Theorem(2.3) show that large enough lower bounds on Size and Depth provide superlinear lower

bounds on Time and Space.

The next theorem presents general relationships between the important network complexity measures.

Theorem 2.4: $\forall f \in B_n$

$$\mathbf{C}_\Omega(f) \leq \mathbf{L}_\Omega(f) < 2^{\mathbf{D}_\Omega(f)}$$

Proof: The first inequality is immediate since formulae are a restricted type of network. The second is obtained by observing that a formula of depth $\mathbf{D}_\Omega(f)$ can be obtained from a network of the same depth simply by duplicating each node until none has fanout exceeding 1 and noting that the binary tree which results has less than $2^{\mathbf{D}_\Omega(f)}$ nodes. \square

By considering the function $\bigwedge_{i=1}^n x_i$ it can be seen that these inequalities are the best possible. Inequalities in the other direction are examined in Section(2.3).

We conclude this section by examining reductions between families of Boolean functions. The ideas presented below are developed in Skyum and Valiant (1985).

Definition 2.1: Let $f(\mathbf{X}_n)$ and $g(\mathbf{Y}_p)$ be n -input and p -input Boolean functions, where $p \geq n$. f is a *projection* of g if there exists a mapping $\sigma: \mathbf{Y}_p \rightarrow \{\mathbf{X}_n, \bar{x}_1, \dots, \bar{x}_n, 0, 1\}$ such that $f(\mathbf{X}_n) = g(\sigma(\mathbf{Y}_p))$. If the mapping σ does not contain negated variables in its range, then f is a *monotone projection* of g . A family of Boolean functions $[f_n]$ is a projection of another family $[g_n]$ if each $f^{(n)}$ is a projection of some $g^{(p)}$. If H is a set of families, the family $[f_n]$ is *universal* for H if every family in H is a projection of $[f_n]$. •

Projections between families provide a natural and precise framework for investigating reducibility between Boolean functions.

The concept of universality, as defined above, requires some strengthening to cater for ideas of computationally efficient reductions. p -projections provide one method of achieving this. A family $[f_n]$ is a p -projection of a family $[g_n]$ if there exists a polynomial q , such that each $f^{(n)}$ is a projection of $g^{(p)}$ for some $p \leq q(n)$. Using this notion a stronger form of universality, p -universality, can be defined in the obvious way. As was observed in (Skyum & Valiant, 1985), the fact that a family $[f_n]$ is p -universal for a class H may provide one mechanism for resolving a number of open problems concerning the complexity of specific functions and the relation between various complexity classes. For example the following non-uniform analogue of the class NP is introduced.

Definition 2.2: Let $f(\mathbf{X}_n)$ and $g(\mathbf{Y}_p)$ be as Defn(2.1). g defines f if and only if:

$$f(\mathbf{X}_n) = \bigvee_{\alpha \in \{0,1\}^{p-n}} g(\mathbf{X}_n, \alpha)$$

This reflects the idea of "searching" through a, possibly exponential, number of choices. A family $[f_n]$ is said to be p -definable if for some polynomial $t(n)$, each $f^{(n)}$ is defined by some function g , for which $\mathbf{C}(g) \leq t(n)$. pD will denote the class of all p -definable families. A family $[f_n]$ is p -complete for a class of families, H , if and only if $[f_n]$ is in H and is p -universal for H . •

Informally, just as pC , the class of families which have polynomial combinational complexity, can be seen as a non-uniform version of P , so the class pD may be interpreted as a non-uniform analogue of NP . Since any family trivially defines itself one has immediately $pC \subseteq pD$. From Corollary(2.1) it may be shown that to separate the classes pC and pD is to separate P and NP , although the converse may not be true. The families p -complete for pD form a core of

problems which are the "hardest" in pD , in the sense that if any one of these were in pC then $pC = pD$. The following elegant result from (Skyum & Valiant, 1985) identifies a specific p -complete family in pD and is similar in spirit to Cook's Theorem (Cook, 1971).

Theorem 2.5: Let $\mathbf{X}_{n,n}$, $\mathbf{Y}_{n,n}$ be disjoint sets of n^2 Boolean variables. $SAT(\mathbf{X}_{n,n}, \mathbf{Y}_{n,n})$ is the Boolean function which is 1 if and only the CNF over $\mathbf{Z} = \{z_1, z_2, \dots, z_n\}$ defined from assignments α to $\mathbf{X}_{n,n}$, β to $\mathbf{Y}_{n,n}$ by:

$$R_{2n^2+n}(\alpha, \beta, \mathbf{Z}) = \bigwedge_{i=1}^n \left(\bigvee_{j=1}^n \alpha_{ij} z_j \vee \beta_{ij} \bar{z}_j \right)$$

has a satisfying assignment for some $\gamma \in \{0, 1\}^n$.

$SAT(\mathbf{X}_{n,n}, \mathbf{Y}_{n,n})$ is p -complete for pD .

Proof: Omitted. \square

A family of functions can be shown to be p -complete for pD by exhibiting a p -projection from a known p -complete family for pD ; it may be observed that many of the classical polynomial reductions used to prove certain decision problems NP -complete can be adapted with little difficulty to yield p -projections from the corresponding pD family. In Chapter(3) some examples of p -projections between monotone Boolean functions will be presented; these are of interest in deriving lower bounds for particular families and as a means of reformulating the question $P = ? NP$. Non-uniform analogues of $DLOGSPACE$ and $PSPACE$ may also be defined.

2.2) Complexity Results For Almost All Boolean Functions

The results proved in Sect(2.1) establish that Ω -networks, for any complete $\Omega \subseteq B_2$, are a reasonable computational model in that lower bounds on Size and Depth, of sufficient magnitude, imply non-

trivial lower bounds on Time and Space. As we have observed previously, the non-uniform nature of this model means that upper bounds on Size and Depth do not necessarily permit corresponding bounds on Time and Space to be inferred. Thus to strengthen the assertion that Boolean networks *are* a reasonable model we should consider the questions below:

B1) What is the asymptotic value of

$$\mathbf{C}(B_n) = \max \{ \mathbf{C}(f) : f \in B_n \}$$

B2) What is the asymptotic value of

$$\mathbf{D}(B_n) = \max \{ \mathbf{D}(f) : f \in B_n \}$$

Obviously if $\mathbf{C}(B_n) = O(n^k)$ for some fixed k , then combinational networks would be an inappropriate model in which to attempt to resolve the question $P = ? NP$.

This section is mainly devoted to determining these quantities. A lower bound on $\mathbf{C}(B_n)$ is given by a result of Shannon (1949). The counting argument introduced there is central to many other similar lower bound proofs. A matching upper bound is proved using the methods of Lupanov (1958). A lower bound on $\mathbf{D}(B_n)$ is deduced from a lower bound on $\mathbf{L}(B_n)$ the analogously defined measure for formulae, using Theorem(2.4). A matching upper bound is provided by the construction of Gaskov (1978) improving the uniform method of McColl (1976), McColl and Paterson (1977)

An important property of all these lower bound results is that they hold for "almost all" f in B_n , i.e

Suppose Π is a property of Boolean functions and $P(n)$ denotes:

$$|\{ f \in B_n : \Pi \text{ is true of } f \}|$$

Π is said to hold for almost all f in B_n if:

$$\lim_{n \rightarrow \infty} \frac{P(n)}{|B_n|} = 1$$

Theorem 2.6: (Shannon, 1949) $\forall \varepsilon > 0$ and n sufficiently large. For almost all $f \in B_n$,

$$C(f) > \frac{(1 - \varepsilon)2^n}{n}$$

Proof: We proceed by estimating the number of distinct optimal n -input networks containing at most M gates. Since each such network can be minimal for at most one Boolean function we can then argue that if $M \leq \frac{(1 - \varepsilon)2^n}{n}$ for any $\varepsilon > 0$, then the number of distinct networks available is only $o(|B_n|)$, and this will be sufficient to prove the theorem.

To start an upper bound on the number of such networks containing exactly m gates is obtained. Let the gates be numbered $1, 2, \dots, m$. Any network can be completely specified by describing for each gate its operation and the two nodes which supply its inputs. From Lemma(1.3)(ii) there are 10 choices of operation for each gate and thus 10^m distinct labellings. The inputs for a gate are either from an input node or from one of the $m - 1$ other gates. So there are no more than $(n + m - 1)^{2m}$ interconnection schemes, giving at most $10^m(n + m - 1)^{2m}$ optimal networks. Now since the gates have been numbered $1, 2, \dots, m$ each distinct optimal network is counted $m!$ times in this analysis. It follows that the number of distinct optimal n -input combinational networks with at most M gates does not exceed:

$$S(M) = \sum_{m=0}^M \frac{10^m(n + m - 1)^{2m}}{m!}$$

$S(M)$ is asymptotically equal to the last term in the summation above. It is easy to verify that if $M \leq \frac{(1-\varepsilon)2^n}{n}$, for any $\varepsilon > 0$, then $S(M)$ is at most $2^{(1-\varepsilon)2^n}$, which is $o(|B_n|)$ as desired. \square

The approach employed in this proof is extremely robust and may be used to derive lower bounds on the complexity of almost all functions for subsets of B_n , such as M_n , when realised by combinational or restricted forms of networks, e.g formulae. Such applications involve the estimation of two quantities: suppose $H_n \subseteq B_n$, which is "well-behaved" in a sense we will not precisely specify (however all specific choices of H_n , such as M_n , which are examined subsequently will be "well-behaved"). Further suppose that A is a class of Boolean networks, which can realise any function in H_n . For a network T in A , $A(T)$ will denote the number of gates in T ; $A(f)$, for $f \in H_n$, will denote:

$$\min \{ A(T) : T \in A \text{ and } T \text{ realises } f \}$$

Shannon's argument shows that a lower bound on $A(f)$, which holds for almost all $f \in H_n$, can be obtained from:

- B1) A lower bound on $|H_n|$
- B2) An upper bound on $|\{ T : T \in A, A(T) = m \}|$

For example if A is the class of combinational networks, we have:

Corollary 2.2: For almost all $f \in H_n$:

$$C(f) \geq \frac{\log |H_n|}{\log \log |H_n|} \quad \square$$

Lupanov (1958) gives a construction which asymptotically matches Shannon's lower bound so that the multiplicative constant c

in the bound $\frac{c \cdot 2^n}{n}$ cannot be proved to be greater than 1. First we show how to construct a network of size $O(2^n)$ for any n -input Boolean function.

Consider the set of all 2^n products of length n (i.e containing n literals) over the set $\{x_1, \dots, x_n, \bar{x}_1, \dots, \bar{x}_n\}$, so that for each product, p , in this set and each i , exactly one of the literals x_i, \bar{x}_i occurs in p . Each p has the form:

$$p = (x_1)^{a_1} \wedge \dots \wedge (x_n)^{a_n}$$

where $(x_i)^{a_i}$ is the literal x_i if a_i is 1, and the literal \bar{x}_i otherwise.

For two such products p, q , of length $n - i + 1$, over the literal set

$$\{x_i, \dots, x_n, \bar{x}_i, \dots, \bar{x}_n\}$$

we define a lexicographic ordering \leq_L by:

$$p \leq_L q \iff$$

$$(p = q) \text{ or}$$

$$(p = \bar{x}_i \wedge p' \text{ and } q = x_i \wedge q') \text{ or}$$

$$(p = (x_i)^a \wedge p' \text{ and } q = (x_i)^a \wedge q' \text{ and } p' \leq_L q').$$

where p', q' are products not involving x_i or \bar{x}_i

In this ordering no product is $\leq_L \bar{x}_1 \cdots \bar{x}_n$ and every product is $\leq_L x_1 \cdots x_n$.

The n -ordered network \mathbf{U}_n is a combinational network with inputs $\{\mathbf{X}_n, \bar{x}_1, \dots, \bar{x}_n\}$, 2^n outputs $\langle t_1, \dots, t_{2^n} \rangle$ for which $res(t_i)$ is the i 'th product in the \leq_L ordering (so $res(t_1) = \bar{x}_1 \cdots \bar{x}_n$).

\mathbf{U}_n is inductively defined as follows:

- U1) If $n=0$ then \mathbf{U}_n is a single node computing the constant function 1.
- U2) If $n>0$ then \mathbf{U}_n is formed by adding 2 new \wedge -gates, t_{2i-1} , t_{2i} and wires

$$\langle \bar{x}_n, t_{2i-1} \rangle ; \langle t_i, t_{2i-1} \rangle ; \langle t_i, t_{2i} \rangle ; \langle x_n, t_{2i} \rangle$$

to \mathbf{U}_{n-1} for each $1 \leq i \leq 2^{n-1}$.

Clearly the node computing 1 and the \wedge -gates of \mathbf{U}_n form a complete binary tree with 2^n leaves and thus contains $2^{n+1} - 1$ gates. Together with the n gates to compute $\{\bar{x}_i \mid 1 \leq i \leq n\}$ this gives:

$$C(\mathbf{U}_n) = 2^{n+1} + n - 1 \leq (1 + \varepsilon) 2^{n+1} \quad \forall \varepsilon > 0$$

Any f could be realised from \mathbf{U}_n just by \vee -ing all the outputs which correspond to satisfying assignments of f , but the resulting network would contain too many gates. The solution adopted in Lupanov (1958) optimises \mathbf{U}_n by employing an expansion of f which has become known as the *Lupanov decomposition*.

An important function which is used frequently below is the *equivalence function*, $\delta_\alpha(\mathbf{X}_n)$, which tests if an assignment to \mathbf{X}_n is exactly the same as α in $\{0, 1\}^n$. Formally, for $\alpha = \langle a_1, \dots, a_n \rangle$:

$$\delta_\alpha(\mathbf{X}_n) = \bigwedge_{i=1}^n (x_i \iff a_i)$$

To simplify the notation we shall use \mathbf{Y} to denote the subset $\{x_1, x_2, \dots, x_k\}$ of \mathbf{X}_n and \mathbf{Z} to denote $\mathbf{X}_n - \mathbf{Y}$, so that $f(\mathbf{X}_n) = f(\mathbf{Y}, \mathbf{Z})$.

Let f be any Boolean function. $\forall 1 \leq k \leq n$, $f(\mathbf{Y}, \mathbf{Z})$ may be defined by a Boolean matrix, $M(f)$, having 2^k rows and 2^{n-k} columns. Each row is labelled with a distinct member of $\{0, 1\}^k$, corresponding to an assignment to \mathbf{Y} . Similarly each column is labelled

with a distinct member of $\{0, 1\}^{n-k}$ corresponding to an assignment to \mathbf{Z} . For the i 'th row let α_i be the associated assignment and let β_j be that which labels the j 'th column. The (i, j) entry of $M(f)$ is then just the value of $f(\alpha_i, \beta_j)$. It will be convenient to refer to row/columns by the assignments labelling them.

For some s , to be fixed later, partition the rows of $M(f)$ into d blocks R_1, R_2, \dots, R_d . R_i contains s consecutive rows ($1 \leq i < d$) and R_d contains at most s rows, hence $d \leq \frac{2^k}{s} + 1$. Any such block will be termed a *rod*. For each rod R_i the function, $rod_i(\mathbf{Y}, \mathbf{Z})$ is given by:

$$rod_i(\mathbf{Y}, \mathbf{Z}) = \bigvee_{\alpha \in \{0,1\}^k \cap R_i} \delta_\alpha(\mathbf{Y}) \wedge f(\alpha, \mathbf{Z})$$

i.e. $rod_i(\mathbf{Y}, \mathbf{Z})$ is $f(\alpha, \mathbf{Z})$ if the assignment α to \mathbf{Y} corresponds to some row in R_i , and is 0 otherwise.

Since $\bigcup_{i=1}^d R_i = \{0, 1\}^k$, so $f(\mathbf{Y}, \mathbf{Z}) = \bigvee_{i=1}^d rod_i(\mathbf{Y}, \mathbf{Z})$. Every rod contains at most s rows and exactly 2^{n-k} columns; if k is "small" compared to s then many of the columns of length s in R_i must be identical.

Let $\langle r_{i,1}, r_{i,2}, \dots, r_{i,s} \rangle$ be the rows in the rod R_i . For $\mathbf{v} = \langle v_1, \dots, v_s \rangle$ in $\{0, 1\}^s$, the (i, \mathbf{v}) -pillar of M , denoted $P_{i,\mathbf{v}}$ is the set of columns in $M(f)$ which satisfy:

$$p \in P_{i,\mathbf{v}} \iff \forall 1 \leq q \leq s \ (r_{i,q}, p) = v_q$$

(Informally $P_{i,\mathbf{v}}$ is the set of columns which when intersected with R_i yield the same s -tuple, \mathbf{v} .)

Note that if \mathbf{v} and \mathbf{u} are different elements of $\{0, 1\}^s$ then the sets $P_{i,\mathbf{u}}$ and $P_{i,\mathbf{v}}$ are disjoint. We can now define a function

$pillar_{i,\mathbf{v}}(\mathbf{Y}, \mathbf{Z})$ by:

$$pillar_{i,\mathbf{v}}(\mathbf{Y}, \mathbf{Z}) = \bigvee_{\beta \in \{0,1\}^{n-k} \cap P_{i,\mathbf{v}}} \delta_{\beta}(\mathbf{Z}) \wedge rod_i(\mathbf{Y}, \beta)$$

Thus:

$$rod_i(\mathbf{Y}, \mathbf{Z}) = \bigvee_{\mathbf{v} \neq \mathbf{0}} pillar_{i,\mathbf{v}}(\mathbf{Y}, \mathbf{Z})$$

(where $\mathbf{0} = \langle 0, 0, \dots, 0 \rangle$)

Now $pillar_{i,\mathbf{v}}(\mathbf{Y}, \mathbf{Z})$ can only equal 1 when the assignment to (\mathbf{Y}, \mathbf{Z}) selects both a column in $P_{i,\mathbf{v}}$ (using \mathbf{Z}) and a row in R_i (using \mathbf{Y}) whose intersection in $M(f)$ equals 1. We can thus express $pillar_{i,\mathbf{v}}(\mathbf{Y}, \mathbf{Z})$ as the conjunction of a function over \mathbf{Y} and another function over \mathbf{Z} , viz:

$$pillar_{i,\mathbf{v}}(\mathbf{Y}, \mathbf{Z}) = row - match_{i,\mathbf{v}}(\mathbf{Y}) \wedge col - match_{i,\mathbf{v}}(\mathbf{Z})$$

where

$$col - match_{i,\mathbf{v}}(\mathbf{Z}) = \bigvee_{\beta \in \{0,1\}^{n-k} \cap P_{i,\mathbf{v}}} \delta_{\beta}(\mathbf{Z})$$

$$row - match_{i,\mathbf{v}}(\mathbf{Y}) = \bigvee_{\alpha \in \{0,1\}^k \cap R_i} \delta_{\alpha}(\mathbf{Y}) \wedge v(\alpha)$$

$v(\alpha)$ being the point within \mathbf{v} selected by the row labelled α in R_i , so if α is the row $r_{i,q}$, then $v(\alpha) = v_q$.

In summary $f(\mathbf{Y}, \mathbf{Z})$ may be written as:

$$\begin{aligned} \bigvee_{i=1}^d rod_i(\mathbf{Y}, \mathbf{Z}) &= \bigvee_{i=1}^d \left(\bigvee_{\mathbf{v} \neq \mathbf{0}} pillar_{i,\mathbf{v}}(\mathbf{Y}, \mathbf{Z}) \right) \\ &= \bigvee_{i=1}^d \left(\bigvee_{\mathbf{v} \neq \mathbf{0}} [row - match_{i,\mathbf{v}}(\mathbf{Y}) \wedge col - match_{i,\mathbf{v}}(\mathbf{Z})] \right) \end{aligned}$$

and this is the (k, s) -Lupanov decomposition of $f(\mathbf{X}_n)$.

Theorem 2.7: (Lupanov, 1958) $\forall \varepsilon > 0$ and n sufficiently large:

$$\forall f \in B_n \quad \mathbf{C}(f) < \frac{(1 + \varepsilon)2^n}{n}$$

Proof: Let $f \in B_n$ and consider the (k, s) -Lupanov decomposition of $f(\mathbf{Y}, \mathbf{Z})$. k and s will be fixed subsequently to obtain the desired bound. From this expansion we can build a network realising f in the following stages:

- L1) Construct the k -ordered network \mathbf{U}_k with inputs \mathbf{Y} and the $(n - k)$ -ordered network \mathbf{U}_{n-k} with inputs \mathbf{Z} . These contain no more than $(1 + \varepsilon)(2^{k+1} + 2^{n-k+1})$ gates in total.
- L2) For each i, \mathbf{v} compute $row - match_{i, \mathbf{v}}(\mathbf{Y})$ by \vee -ing together the appropriate outputs of \mathbf{U}_k . Since $row - match_{i, \mathbf{v}}$ has no more than s satisfying assignments it can be computed with at most $s - 1$ additional gates, giving no more than $d. s. (2^s - 1) \leq 2^{k+s}$ extra gates to compute all of them.
- L3) Similarly compute all the $col - match_{i, \mathbf{v}}(\mathbf{Z})$ by \vee -ing the appropriate outputs of \mathbf{U}_{n-k} . Since these correspond to columns in $P_{i, \mathbf{v}}$, for each i every output of \mathbf{U}_{n-k} is used at most once (recall that if $\mathbf{v} \neq \mathbf{u}$ then $P_{i, \mathbf{v}}$ and $P_{i, \mathbf{u}}$ are disjoint). Thus with i fixed all the $col - match_{i, \mathbf{v}}$ can be computed with an additional 2^{n-k} gates and therefore $\leq d. 2^{n-k}$ over all.
- L4) Conjoin every $row - match_{i, \mathbf{v}}(\mathbf{Y})$ to its correspondent $col - match_{i, \mathbf{v}}(\mathbf{Z})$, using at most $d. 2^s$ gates.
- L5) Finally compute $f(\mathbf{X}_n)$ by \vee -ing all the functions computed in stage (L4). This adds at most a further $d. 2^s$ gates.

Recalling that $d \leq \frac{2^k}{s} + 1$ and summing each contribution, the reader may easily verify that this gives:

$$\mathbf{C}(f) \leq (1 + \varepsilon)(2^{k+1} + 2^{n-k+2} + 2^{s+1}) + 2^{k+s} \left(1 + \frac{2}{s}\right) + \frac{2^n}{s}$$

Choosing $k = \lceil 3 \log n \rceil$ and $s = \lceil n - 5 \log n \rceil$ leaves:

$$\mathbf{C}(f) \leq \frac{2^n}{n - 5 \log n} + o\left(\frac{2^n}{n}\right)$$

Thus for all $\varepsilon > 0$ and sufficiently large n :

$$\mathbf{C}(f) < \frac{(1 + \varepsilon)2^n}{n} \quad \square$$

The lower bound on $\mathbf{D}(B_n)$ is a consequence of the following lower bound on the complexity of almost all n -input formulae over B_2 .

Theorem 2.8: (Riordan and Shannon, 1942) For almost all $f \in B_n$

$$\mathbf{L}(f) > \frac{2^n}{\log n}$$

Proof: See Chapter(4), Theorem(4.1). \square

Corollary 2.3: For all $\varepsilon > 0$ and sufficiently large n . For almost all $f \in B_n$:

$$\mathbf{D}(f) > n - \log \log n$$

Proof: Immediate from Thm(2.4), using $\mathbf{L}(f) \leq 2^{\mathbf{D}(f)}$, and Thm(2.8). \square

The upper bound of Gaskov (1978) which matches this lower bound to within an additive constant is based on a construction of Lupanov (1973). This method is non-uniform unlike the earlier upper bound McColl and Paterson (1977), which showed $\mathbf{D}(B_n) \leq n + 1$, and was built around the notion of formula schemes. Below we describe

both constructions.

Definition 2.3: A *formula scheme* is a directed acyclic graph in which nodes have in-degree 2 (gates) or 0 (inputs). All gates have out-degree at most 1. Let $H_n \subseteq B_n$. A formula scheme, C_n , covers H_n over the basis $\Omega \subseteq B_2$ if and only if for each $f \in H_n$ the gates of C_n can be assigned operations in Ω so that the resulting formula realises f . •

We shall construct a formula scheme which covers B_n over the basis B_2 and has depth $n + 1$. As regards schemes this depth bound is very close to optimal; an easy counting argument can be used to show that any formula scheme with the required property must have depth at least $n - 1$.

In describing the construction the correspondence between Ω -expressions and formulae, as given in Defn(1.3), will be used. It should be clear throughout that the method presented *does* describe a scheme.

Let $\mathbf{Y} = \langle y_1, \dots, y_k \rangle$ and $\mathbf{Z} = \langle z_1, \dots, z_m \rangle$ be disjoint sets of Boolean variables and $f(\mathbf{Y}, \mathbf{Z}) \in B_{k+m}$. The *disjunctive expansion* of $f(\mathbf{Y}, \mathbf{Z})$ about \mathbf{Z} is given by:

$$f(\mathbf{Y}, \mathbf{Z}) = \bigvee_{\alpha \in \{0,1\}^m} \delta_\alpha(\mathbf{Z}) \wedge f(\mathbf{Y}, \alpha)$$

The dual, *conjunctive expansion* of $f(\mathbf{Y}, \mathbf{Z})$ about \mathbf{Z} , is defined to be:

$$f(\mathbf{Y}, \mathbf{Z}) = \bigwedge_{\alpha \in \{0,1\}^m} (\bar{\delta}_\alpha(\mathbf{Z}) \vee f(\mathbf{Y}, \alpha))$$

where $\bar{\delta}_\alpha(\mathbf{Z})$ denotes the complement of $\delta_\alpha(\mathbf{Z})$.

Suppose that, following Spira (1971b), one attempts to build a formula of minimal depth for $f(\mathbf{Y}, \mathbf{Z})$ using these expansions as the vehicle for a recursive construction. Each δ_α , $\bar{\delta}_\alpha$ term being in effect

a product or sum of m literals (since $\alpha \in \{0, 1\}^m$) it can be realised in depth $\lceil \log m \rceil$. To compute $f(\mathbf{Y}, \mathbf{Z})$, formulae for each subterm δ_α , $f(\mathbf{Y}, \alpha)$ can be constructed in parallel and joined together using one extra level. Finally all the 2^m subterms must be collected, this requiring depth m . Thus the following recurrence relation results:

$$\begin{aligned} \mathbf{D}(f(\mathbf{Y}, \mathbf{Z})) &\leq 1 + m + \max \{ \lceil \log m \rceil, \max_{\alpha \in \{0, 1\}^m} \{\mathbf{D}\} \} \\ &\leq 1 + m + \max \{ \lceil \log m \rceil, \mathbf{D}(g(\mathbf{Y})) \} \end{aligned}$$

for some $g \in B_k$.

By choosing a suitable partition of \mathbf{X}_n , Spira was able to construct a formula scheme covering B_n over B_2 with depth $n + \log^* n$. The $\log^* n$ arises through the additional level of gates, used to pair the δ_α , $\bar{\delta}_\alpha$ to their corresponding $f(\mathbf{Y}, \alpha)$ terms, at each recursive step.

The scheme developed in (McColl and Paterson, 1977) eliminates these levels by applying an ingenious optimisation to the basic recursive construction.

Let $\delta_\alpha(\mathbf{Z}) \wedge f(\mathbf{Y}, \alpha)$ be a single term of the disjunctive expansion about \mathbf{Z} . For any $\mathbf{W} \subseteq \mathbf{Y}$ we may express $f(\mathbf{Y}, \alpha)$ as a product of $2^{|\mathbf{W}|}$ terms by using the conjunctive expansion of $f(\mathbf{Y}, \alpha)$ about \mathbf{W} .

$$\delta_\alpha(\mathbf{Z}) \wedge f(\mathbf{Y}, \alpha) = \delta_\alpha(\mathbf{Z}) \wedge \bigwedge_{\beta \in \{0, 1\}^{|\mathbf{W}|}} [\bar{\delta}_\beta(\mathbf{W}) \vee f(\mathbf{U}, \beta, \alpha)]$$

where $\mathbf{U} = \mathbf{Y} - \mathbf{W}$.

Unfortunately this is a product of $2^{|\mathbf{W}|} + 1$ terms and would require additional depth of $|\mathbf{W}| + 1$, as before. The novel solution adopted is to discard one of the subterms $[\bar{\delta}_\beta(\mathbf{W}) \vee f(\mathbf{U}, \beta, \alpha)]$, leaving a product of $2^{|\mathbf{W}|}$ terms, realisable with additional depth only $|\mathbf{W}|$. By alternating disjunctive and conjunctive expansions in the recursion

this process of term disposal may be employed at each stage to guarantee that the number of terms in a sum or product is an exact power of 2.

Of course the resulting formula will not realise $f(\mathbf{Y}, \mathbf{Z})$ but will merely be an "approximation" to it, having depth n . The final stage is a remarkable result which shows that $f(\mathbf{Y}, \mathbf{Z})$ can be recovered from the approximation, f^* , simply by \oplus -ing f^* with a suitable "correcting" function, R , also constructed in depth n by using the method recursively. The depth of the final scheme will be:

$$\mathbf{D}(f(\mathbf{Y}, \mathbf{Z})) \leq \max \{\mathbf{D}(f^*), \mathbf{D}(R)\} + 1 \leq n + 1$$

We need to define a partition of $\{R_0, R_1, \dots, R_p\}$ of \mathbf{X}_n . The actual elements in $\mathbf{X}_n \cap R_i$ are not important, however the relative sizes of the partition components must satisfy several criteria. For $0 \leq i \leq p$, let r_i denote $|R_i|$. For our purpose, any sequence $\langle r_0, r_1, \dots, r_p \rangle \in \{\mathbf{N}\}^{p+1}$ of component sizes which meets the following is suitable.

$$\sum_{j=0}^i r_j \text{ will be denoted by } S_i.$$

$$\text{G1) } r_0 = r_1 = 2$$

$$\text{G2) } S_p = n$$

$$\text{G3) } r_m \leq 2^{S_{m-2}} \text{ for } m \text{ even and } \geq 2$$

$$\text{G4) } r_m \leq 2^{S_{m-2}} - 2^{S_{m-3}} \text{ for } m \text{ odd and } \geq 3$$

The particular sequence employed is defined by the rules:

Seq1)

$$r_0 = 2$$

Seq2)

$$r_i = i + 1 \quad 0 < i < p$$

Seq3)

$$r_p = n - S_{p-1}$$

$$\text{where } p = \max \left\{ q \in \mathbf{N} : 1 + \frac{q(q+1)}{2} < n \right\}$$

The choice of sequence affects the additive term for the depth of schemes over bases other than B_2 .

Definition 2.4: Let $S = \{R_1, R_2, \dots, R_k\}$ where $R_i \subseteq \mathbf{X}_n \forall 1 \leq i \leq k$. A Boolean function $g(\mathbf{X}_n)$ is S -simple if $g(\mathbf{X}_n) = 0$ whenever $R_i = \mathbf{0}$ for any $1 \leq i \leq k$, ($\mathbf{0}$ denotes $\langle 0, 0, \dots, 0 \rangle$). •

Note that S is not required to be a partition of \mathbf{X}_n or to consist of disjoint subsets.

Lemma 2.1: Let $\{R_0, R_1, \dots, R_m\}$ (where $m \geq 1$) be disjoint subsets of \mathbf{X}_n whose cardinalities $\langle r_0, r_1, \dots, r_m \rangle$ satisfy conditions (G1), (G3) and (G4) above. If $g(R_0, \dots, R_m)$ is $\{R_1, \dots, R_m\}$ -simple then there is a formula for g which is:

Case a: (m odd) A disjunction of $2^{r_m} - 1$ subformulae, each of depth S_{m-1} .

Case b: (m even) A conjunction of $2^{r_m} - 1$ subformulae, each of depth S_{m-1} , and one additional subformula of depth S_{m-2} .

Proof: By induction on m . Consider the two possible expansions of $g(R_0, \dots, R_m)$ about R_m . In Case(a) we have:

$$g(R_0, \dots, R_m) = \bigvee_{\alpha \neq \mathbf{0}} \delta_\alpha(R_m) \wedge g(R_0, \dots, R_{m-1}, \alpha)$$

and in Case(b) $g(R_0, \dots, R_m)$ is equal to,

$$\bar{\delta}_0(R_m) \wedge \bigwedge_{\alpha \neq \mathbf{0}} [\bar{\delta}_\alpha(R_m) \vee g(R_0, \dots, R_{m-1}, \alpha)]$$

If $m = 1$, Case(a) applies and then from (G1):

$$g(R_0, R_1) = \bigvee_{\alpha \neq 0} \delta_\alpha(R_1) \wedge g(R_0, \alpha)$$

and this is a disjunction of 3 subformulae each of which is realisable by a formula of depth 2 over the basis B_2 , since, again using (G1), the terms $\delta_\alpha(R_1)$, $g(R_0, \alpha)$ are both in B_2 .

This establishes the inductive base. Now suppose the lemma holds for all values $< m$ and let $m > 1$ be odd. From the inductive hypothesis, the component $g(R_0, \dots, R_{m-1}, \alpha)$ of Case(a), may be expressed as a conjunction of $2^{r_{m-1}} - 1$ subformulae, each of depth S_{m-2} and an additional subformula of depth S_{m-3} . Thus $g(R_0, \dots, R_m)$ is equal to

$$\bigvee_{\alpha \neq 0} \delta_\alpha(R_m) \wedge \bar{\delta}_0(R_{m-1}) \wedge \bigwedge_{\beta \neq 0} [\bar{\delta}_\beta(R_{m-1}) \vee g(R_0, \dots, R_{m-2}, \beta, \alpha)]$$

Now $\delta_\alpha(R_m)$ is effectively a product of r_m literals, and $\bar{\delta}_0(R_{m-1})$ can be realised by a formula of depth S_{m-3} . From (G4) $r_m \leq 2^{S_{m-2}} - 2^{S_{m-3}}$, so the sub-expression $\delta_\alpha(R_m) \wedge \bar{\delta}_0(R_{m-1})$ can be computed by a formula of depth S_{m-2} .

From the inductive hypothesis the sub-expression

$$\bigwedge_{\beta \neq 0} [\bar{\delta}_\beta(R_{m-1}) \vee g(R_0, \dots, R_{m-2}, \beta, \alpha)]$$

is a conjunction of $2^{r_{m-1}} - 1$ subformulae each realisable in depth S_{m-2} , thus the complete term can be computed by a formula of depth $r_{m-1} + S_{m-2} = S_{m-1}$, which proves the inductive step for the case where m is odd.

The case of m being even follows almost directly from the second expansion after observing that the $\bar{\delta}_\alpha$ terms can be realised in depth S_{m-2} . \square

The importance of this lemma is the part it plays in the proof of the next result.

Theorem 2.9: $\forall n > 4$ if $g(\mathbf{X}_n)$ is $\{R_1, R_2, \dots, R_{p-1}\}$ -simple (where the R_i have cardinalities agreeing with the sequence $\langle r_i \rangle$ given by Seq1-Seq3 above), then there is a formula realising $g(\mathbf{X}_n)$ which has depth n .

Proof: Since $n > 4$ it follows that $p > 1$. Express $g(\mathbf{X}_n)$ as a disjunctive (resp. conjunctive) expansion about R_p if p is odd (resp. even). From Lemma(2.1) each of the 2^{r_p} components of the expansion can be realised by a formula of depth S_{p-1} . Thus $g(\mathbf{X}_n)$ can be realised by a formula having depth $S_{p-1} + r_p = n$. \square

The approximation to $f(\mathbf{X}_n)$, as described in the proof outline, will be some $\{R_1, \dots, R_{p-1}\}$ -simple function $g(\mathbf{X}_n)$, which Thm(2.9) shows can be realised in depth n . In order to achieve the desired depth bound of $n + 1$ for any f we must show how to recover f from its simple approximation g .

Lemma 2.2: Let $\{R_1, \dots, R_k\}$ be disjoint subsets of \mathbf{X}_n . $\forall f(\mathbf{X}_n)$ there exist functions:

$$f_1(\mathbf{X}_n - R_1), \dots, f_i(\mathbf{X}_n - R_i), \dots, f_k(\mathbf{X}_n - R_k)$$

such that the function

$$g(\mathbf{X}_n) = f(\mathbf{X}_n) \oplus \bigoplus_{i=1}^k f_i(\mathbf{X}_n - R_i)$$

is $\{R_1, \dots, R_k\}$ -simple.

Proof: (By induction on $k \geq 1$)

Inductive Base: $k=1$

Define $f_1(\mathbf{X}_n - R_1)$ to be the function

$$f^{|R_1=0}(\mathbf{X}_n - R_1)$$

then certainly $g(\mathbf{X}_n) = f(\mathbf{X}_n) \oplus f_1(\mathbf{X}_n - R_1)$ is $\{R_1\}$ -simple.

Inductive Step: Suppose the result holds for $k-1$ and so there exist $f_1(\mathbf{X}_n - R_1), \dots, f_{k-1}(\mathbf{X}_n - R_{k-1})$ such that $\forall j, 1 \leq j \leq k-1,$

$$R_j = \mathbf{0} \Rightarrow f \oplus \bigoplus_{i=1}^{k-1} f_i = 0$$

Define $f_k(\mathbf{X}_n - R_k)$ by:

$$f_k(\mathbf{X}_n - R_k) = \begin{cases} 0 & \text{if some } R_i = \mathbf{0} \ (1 \leq i \leq k-1) \\ g_{k-1}^{|R_k=0}(\mathbf{X}_n - R_k) & \text{otherwise} \end{cases}$$

Clearly f_k has the desired property. \square

Theorem 2.10: For all $n \geq 1$ there is a formula scheme of depth $n+1$ which covers B_n over basis B_2 .

Proof: For $n \leq 4$ schemes may be constructed directly. For $n > 4,$ using Lemma(2.2) and the properties of \oplus we may express any function $f(\mathbf{X}_n)$ as:

$$g(\mathbf{X}_n) \oplus \bigoplus_{i=1}^{p-1} f_i(\mathbf{X}_n - R_i)$$

where $g(\mathbf{X}_n)$ is $\{R_1, \dots, R_{p-1}\}$ -simple, and the R_i are as above. Using Thm(2.9) yields a formula of depth n for $g(\mathbf{X}_n),$ to this we must \oplus functions f_1, \dots, f_{p-1} to obtain $f(\mathbf{X}_n),$ each f_i having $n-i-1$ arguments. When $n-i-1 \leq 4$ a formula for f_i may be constructed immediately, otherwise we can apply the construction recursively to yield a formula for f_i of depth $n-i.$ We may rearrange, using the associativity of $\oplus,$ the expression for $f(\mathbf{X}_n)$ at the start of

this proof, to obtain:

$$f = g \oplus (f_1 \oplus (f_2 \oplus (\dots \oplus f_{p-1}))) \dots$$

and this represents a formula of depth $n + 1$ since each f_i has depth $n - i$, for $1 \leq i \leq p - 1$. \square

Preparata and Muller (1971) examines $\mathbf{D}(f(\mathbf{X}_n))$ for $n \leq 8$ and proves an upper bound of n on depth for these cases. By computer analysis, based on ideas of Knuth, (Elspas et alia, 1968) shows that this is the best possible for $n = 3, 4$.

Theorem 2.11: (Gaskov, 1978) For all $f \in B_n$,

$$\mathbf{D}(f) \leq \lceil n - \log \log n + o(1) \rceil + 2$$

Proof:^{a)} Let $f \in B_n$ be any Boolean function and partition its inputs \mathbf{X}_n into 4 sets, \mathbf{W} , \mathbf{Y} , \mathbf{Z} and \mathbf{U} of sizes w , y , z and u respectively so that $w + y + z + u = n$ and u is an exact power of 2. The precise values of these quantities is given below. We describe an expansion of $f(\mathbf{W}, \mathbf{Y}, \mathbf{Z}, \mathbf{U})$ whose depth will be of the required order.

Given $\alpha = \langle a_1, \dots, a_u \rangle \in \{0, 1\}^u$, the *sphere with centre α* is the set of u -tuples, $sph(\alpha)$, defined as

$$\bigcup_{i=1}^u \langle a_1, \dots, a_{i-1}, \bar{a}_i, a_{i+1}, \dots, a_u \rangle$$

Now since u is chosen as a power of two it follows that one may choose $2^u/u$ centres, $\alpha^{(1)}, \dots, \alpha^{(2^u/u)}$ whose spheres afford a partition of $\{0, 1\}^u$. For some such choice of centres let $\phi_i(\mathbf{U})$ be the characteristic function of the i 'th sphere, where $1 \leq i \leq 2^u/u$. Clearly

$$\phi_i(\mathbf{U}) \wedge u_l^{-a_l^{(i)}} = \delta_{\alpha^{(i)}}(\mathbf{U})$$

a) The expansion of f which is central to this proof is originally from Lupanov (1973)

where $\alpha_l^{(i)}$ is the tuple in $sph(\alpha^{(i)})$ which differs in its l 'th entry from $\alpha^{(i)}$.

Now let

$$f_{i,\sigma,\rho}(\mathbf{Z}, \mathbf{U}) = \phi_i(\mathbf{U}) f(\sigma, \rho, \mathbf{Z}, \mathbf{U})$$

It is easy to see that $f(\mathbf{W}, \mathbf{Y}, \mathbf{U}, \mathbf{Z})$ is equal to,

$$\bigvee_{\sigma} \bigvee_{\rho} \bigvee_{i=1}^{2^{u/u}} \delta_{\sigma}(\mathbf{W}) \delta_{\rho}(\mathbf{Y}) f_{i,\sigma,\rho}(\mathbf{Z}, \mathbf{U})$$

Consider the function $f_{i,\sigma,\rho}(\mathbf{Z}, \mathbf{U})$ for fixed i , σ and ρ . This may be written as a table with 2^z rows and u columns; the rows representing all possible assignments to \mathbf{Z} and the columns all u -tuples in $sph(\alpha^{(i)})$. The table entry corresponding to row $\zeta \in \{0, 1\}^z$ and column l contains the value of $f_{i,\sigma,\rho}(\zeta, \alpha_l^{(i)})$. Using this table we can express $f_{i,\sigma,\rho}(\mathbf{Z}, \mathbf{U})$ using a variant of the (k, s) -Lupanov representation as follows.

For some $s \leq 2^z$, to be fixed subsequently, partition the rows into $\lfloor 2^z/s \rfloor$ rods, each containing exactly s consecutive rows, and at most one rod of fewer than s rows. Let A_1, \dots, A_p denote these, where $p \leq 2^z/s + 1$. For each $1 \leq k \leq p$ define the function $f_{i,\sigma,\rho,k}(\mathbf{Z}, \mathbf{U})$ to be,

$$f_{i,\sigma,\rho,k}(\mathbf{Z}, \mathbf{U}) = \bigvee_{\zeta \in A_k} \delta_{\zeta}(\mathbf{Z}) f_{i,\sigma,\rho}(\zeta, \mathbf{U})$$

so that,

$$f_{i,\sigma,\rho}(\mathbf{Z}, \mathbf{U}) = \bigvee_{k=1}^p f_{i,\sigma,\rho,k}(\mathbf{Z}, \mathbf{U})$$

We need also a partition of the columns. For $\tau \in \{0, 1\}^s$ let $B_{i,\sigma,\rho,k,\tau}$ be the set of columns whose intersection with the rod A_k is

the s -tuple τ .

Although $B_{i,\sigma,\rho,k,\tau}$ induces a partition of the columns $\{1, \dots, u\}$ using differing τ , this is not good enough to yield the required bound on depth, since the size of $B_{i,\sigma,\rho,k,\tau}$ may be too large. To overcome this we further partition each set $B_{i,\sigma,\rho,k,\tau}$. For some $q \leq u$, to be fixed subsequently, partition each set $B_{i,\sigma,\rho,k,\tau}$ into $\lceil B_{i,\sigma,\rho,k,\tau}/q \rceil$ subsets of size at most q . We denote a typical resulting subset by $B_{i,\sigma,\rho,k,\tau,m}$ where,

$$1 \leq m \leq \lceil B_{i,\sigma,\rho,k,\tau}/q \rceil$$

For fixed i, σ, ρ and k let N denote the number of distinct subsets $B_{i,\sigma,\rho,k,\tau,m}$, i.e

$$N = \left| \bigcup_{\tau} \bigcup_m \{ B_{i,\sigma,\rho,k,\tau,m} \} \right|$$

Then $N \leq \frac{u}{q} + 2^s$. To see this let $r(d)$ denote

$$|\{ \tau : |B_{i,\sigma,\rho,k,\tau}| = d \}|$$

Clearly $\sum_{d=1}^u r(d) = u$ and $r(d) \neq 0$ for at most 2^s values of d . From the definition of $B_{i,\sigma,\rho,k,\tau,m}$ it follows that,

$$N \leq \sum_{d=1}^u \lceil r(d)/q \rceil \leq \sum_{d=1}^u r(d)/q + 2^s = \frac{u}{q} + 2^s.$$

We can thus renumber the subsets $B_{i,\sigma,\rho,k,\tau,m}$ as $B_{i,\sigma,\rho,k,j}$ where $1 \leq j \leq N$. Each of these sets contains at most q elements of $\{1, \dots, u\}$.

Now $f_{i,\sigma,\rho,k}(\mathbf{Z}, \mathbf{U})$ can be expressed in terms of the conjunction of two functions over disjoint variable sets. Specifically,

$$f_{i,\sigma,\rho,k}(\mathbf{Z}, \mathbf{U}) = \bigvee_{j=1}^N f_{i,\sigma,\rho,k,j}^{(1)}(\mathbf{U}) \wedge f_{i,\sigma,\rho,k,j}^{(2)}(\mathbf{Z})$$

where;

$$\begin{aligned} f_{i,\sigma,\rho,k,j}^{(1)}(\mathbf{U}) &= \phi_i(\mathbf{U}) \wedge f_{i,\sigma,\rho,k,j}^{(3)}(\mathbf{U}) \\ &= \phi_i(\mathbf{U}) \wedge \left(\bigvee_{l \in B_{i,\sigma,\rho,k,j}} u_l^{-a_i^{(i)}} \right) \end{aligned}$$

$$f_{i,\sigma,\rho,k,j}^{(2)}(\mathbf{Z}) = \bigvee_{\zeta \in A_k : v(\zeta)=1 \text{ in } \tau} \delta_\zeta(\mathbf{Z})$$

i.e the columns in $B_{i,\sigma,\rho,k,j}$ yield the same z -tuple, τ when intersected with rod A_k . $f^{(2)}$ selects those rows of A_k which indicate entries of τ equal to 1.

In summary, $f(\mathbf{W}, \mathbf{Y}, \mathbf{Z}, \mathbf{U})$ is equal to,

$$\bigvee_{\sigma} \bigvee_{\rho} \bigvee_{i} \bigvee_{k} \bigvee_{j} \delta_\sigma(\mathbf{W}) \delta_\rho(\mathbf{Y}) \phi_i(\mathbf{U}) f_{i,\sigma,\rho,k,j}^{(2)}(\mathbf{Z}) f_{i,\sigma,\rho,k,j}^{(3)}(\mathbf{U})$$

However we need to further rearrange this to yield a suitable expansion. Consider the functions,

$$g_{i,\sigma,k,j}(\mathbf{Y}, \mathbf{Z}) = \bigvee_{\rho} \delta_\rho(\mathbf{Y}) f_{i,\sigma,\rho,k,j}^{(2)}(\mathbf{Z})$$

$$h_{i,\sigma,\rho,k,j}(\mathbf{Y}, \mathbf{U}) = \bigvee_{\rho} \delta_\rho(\mathbf{Y}) f_{i,\sigma,\rho,k,j}^{(3)}(\mathbf{U})$$

It is easy to show that,

$$h_{i,\sigma,\rho,k,j}(\mathbf{Y}, \mathbf{U}) = \bigwedge_{\rho} (\neg \delta_\rho(\mathbf{Y}) \vee f_{i,\sigma,\rho,k,j}^{(3)}(\mathbf{U}))$$

and so,

$$\bigvee_{\rho} \delta_{\rho} f_{i,\sigma,\rho,k,j}^{(2)} f_{i,\sigma,\rho,k,j}^{(3)} = g_{i,\sigma,k,j} \wedge h_{i,\sigma,k,j}$$

With this, the expansion which we analyse the depth of is,

$$\bigvee_{\sigma} \bigvee_i \bigvee_k \bigvee_j \delta_{\sigma}(\mathbf{W}) \phi_i(\mathbf{U}) g_{i,\sigma,k,j}(\mathbf{Y}, \mathbf{Z}) \wedge_{\rho} (\neg \delta_{\rho}(\mathbf{Y}) \bigvee f_{i,\sigma,\rho,k,j}^{(3)}(\mathbf{U})) \quad (2.1)$$

In which: σ ranges over $\{0, 1\}^w$, ρ over $\{0, 1\}^y$, $1 \leq i \leq 2^u/u$, $1 \leq k \leq p$, $1 \leq j \leq N$ and $p = \lceil 2^z/s \rceil$, $N \leq \frac{u}{q} + 2^s$.

We have to fix the values of w , y , z , u , s and q . Set,

$$y = \lfloor 2 \log n \rfloor ; z = \lfloor 2 \log \log n \rfloor ; u = 2^{\lfloor \log n \rfloor - 1}$$

$$s = \lfloor \log n - 5 \log \log n \rfloor$$

and choose q to be in the interval $(\log n)^4 \leq q \leq 3(\log n)^4$ in such a way that $y+q$ is an exact power of 2.

We claim that these choices yield the desired depth bound using our final expansion of $f(\mathbf{W}, \mathbf{Y}, \mathbf{Z}, \mathbf{U})$ above.

Recall that the disjunction or conjunction of r literals can be realised by a network of depth $\lceil \log r \rceil$. To simplify the derivation let $F_{i,\sigma,k,j}^1$ denote the function

$$\delta_{\sigma}(\mathbf{W}) \phi_i(\mathbf{U}) g_{i,\sigma,k,j}(\mathbf{Y}, \mathbf{Z})$$

$F_{i,\sigma,\rho,k,j}^2$ the function,

$$\neg \delta_{\rho}(\mathbf{Y}) \bigvee f_{i,\sigma,\rho,k,j}^{(3)}(\mathbf{U})$$

and $F_{i,\sigma,k,j}^3$ the function $\bigvee_{\rho} F_{i,\sigma,\rho,k,j}^2(\mathbf{Y}, \mathbf{U})$. With these our expansion

is,

$$f = \bigvee_{\sigma} \bigvee_i \bigvee_k \bigvee_j F_{i,\sigma,k,j}^1 \wedge F_{i,\sigma,k,j}^3 \quad (2.2)$$

It follows that,

$$\mathbf{D}(f) \leq \lceil \log \left(\frac{2^u}{u} 2^w p N \right) \rceil + 1 + \max_{i,\sigma,k,j} \{ \mathbf{D}(F^1), \mathbf{D}(F^3) \} \quad (2.3)$$

Consider $\mathbf{D}(F_{i,\sigma,k,j}^1)$. Clearly,

$$\mathbf{D}(\delta_{\sigma}(\mathbf{W})) \leq 2 + \log n \quad ; \quad \mathbf{D}(\phi_i(\mathbf{U})) \leq 2 \log n - 1 \quad (2.4)$$

Also by expressing $g_{i,\sigma,k,j}(\mathbf{Y}, \mathbf{Z})$ in disjunctive normal form we can construct a network for this of depth at most

$$y + z + \lceil \log(y + z) \rceil \leq 2 \log n + 3 \log \log n + 3 + o(1) \quad (2.5)$$

From (2.1), (2.4) and (2.5) it follows that,

$$\mathbf{D}(F_{i,\sigma,j,k}^1) \leq 2 \log n + 3 \log \log n + 5 + o(1) \quad (2.6)$$

Now consider $F_{i,\sigma,\rho,k,j}^2(\mathbf{Y}, \mathbf{U})$. Since $|B_{i,\sigma,\rho,k,j}| \leq q$ and $y + q = 2^r$ for some integral r , we have,

$$\mathbf{D}(F_{i,\sigma,\rho,k,j}^2) \leq 1 + \log(y + q) = r + 1 \quad (2.7)$$

and so from the definition of $F_{i,\sigma,j,k}^3$,

$$\mathbf{D}(F_{i,\sigma,j,k}^3) \leq y + r + 1 \quad (2.8)$$

With (2.6), (2.8) and the choice of y and q it follows that,

$$\max \{ \mathbf{D}(F_{i,\sigma,j,k}^1), \mathbf{D}(F_{i,\sigma,j,k}^3) \} \leq y + r + 1 \quad (2.9)$$

(2.3), (2.8) and (2.9) yield,

$$\mathbf{D}(f) \leq \lceil \left(\frac{2^{w+u}}{u} \right) p N \rceil + y + r + 2 \quad (2.10)$$

As $n \rightarrow \infty$, $2^s = o(u/q)$ and $2^z/s \rightarrow \infty$ hence,

$$\frac{2^{w+u}}{u} p N = \frac{2^{n-y}}{sq} (1 + o(1))$$

So from (2.10)

$$\begin{aligned} \mathbf{D}(f) &\leq \lceil n - y - \log s - \log q + o(1) \rceil + y + r + 2 \\ &= \lceil n - \log s - \log q + m + o(1) \rceil + 2 \\ &= \lceil n - \log s + o(1) \rceil + 2 \\ &= \lceil n - \log \log n + o(1) \rceil + 2 \end{aligned}$$

This proves the upper bound claimed. \square

A natural question to consider at this stage is how large may the "gaps" be for the complexity measures Ω -network size and depth. A gap in this sense is any non-empty interval of natural numbers (c, d) such that no function in B_n has Ω -network size (depth) which is greater than c but less than d .

In order to discuss this question more precisely we introduce the following notation:

For any complexity measure M , let

$$\mathbf{M}(r) = \{f \in B_n \mid M(f) \leq r\}$$

and

$$M(B_n) = \max \{ f \in B_n \mid M(f) \}$$

(where $M = \mathbf{C}_\Omega$ or \mathbf{D}_Ω and Ω is not complete, $M(B_n)$ will be regarded as the maximal complexity of any function in B_n covered by Ω)

The problem of determining the size of complexity gaps can now be formulated as that of calculating:

$$m_\Omega(r) = \min \{ c \in \mathbf{N} : \mathbf{M}(r) \subset \mathbf{M}(r+c) \}$$

Thus we examine questions concerned with complexity hierarchies, particular complexity measures of interest being \mathbf{C}_Ω , \mathbf{D}_Ω and \mathbf{L}_Ω . For these we use c_Ω , d_Ω and l_Ω to denote the corresponding instantiations of m_Ω . The approach taken in the proof of such results is to construct a sequence; f_0, f_1, \dots, f_r of functions in B_n which satisfies:

$$M(f_r) = M(B_n)$$

$$M(f_i) \leq M(f_{i-1}) + b \quad 0 < i \leq r$$

where b depends on the hierarchy sought.

The study of these problems was initiated in (McColl, 1977) and (McColl, 1978b). The results therein have subsequently been improved, however the latter paper proves the existence of a uniform hierarchy for Ω -depth.

Lemma 2.3: (McColl, 1978b) $\forall \Omega \subseteq B_2$;

$$d_\Omega(r) = 1 \quad \forall r < \mathbf{D}_\Omega(B_n)$$

Proof: Let f be any function in B_n for which $\mathbf{D}_\Omega(f) = \mathbf{D}_\Omega(B_n)$. We construct a sequence of functions in B_n :

$$x_i = f_0, f_1, \dots, f_r = f$$

such that $\mathbf{D}_\Omega(f_i) = \mathbf{D}_\Omega(f_{i-1}) + 1$ for each $i \geq 1$. The existence of such a sequence will clearly prove the lemma. So suppose for some $0 \leq i < r$ we have found a sequence of functions in B_n ;

$$f_{i+1}, f_{i+2}, \dots, f_r$$

such that $\mathbf{D}_\Omega(f_r) = \mathbf{D}_\Omega(B_n)$ and $\mathbf{D}_\Omega(f_j) = \mathbf{D}_\Omega(f_{j-1}) + 1$. Consider any minimal depth Ω -network, T , which computes f_{i+1} at some node t . Let v, w be the nodes of T which supply the inputs of t . Certainly

$$\mathbf{D}_\Omega(f_{i+1}) = \max \{ \mathbf{D}_\Omega(\text{res}(v)), \mathbf{D}_\Omega(\text{res}(w)) \} + 1$$

and so one of $\text{res}(v), \text{res}(w)$ is a suitable choice for f_i . \square

One weakness of McColl's hierarchy is that it may contain degenerate members. Wegener (1981) proved that the lemma above still holds with the constraint that only non-degenerate functions are permitted. The proof is rather lengthy and we refer the reader to Wegener's paper for the details of this result.

For Ω -network size the methods of Paterson and Wegener (1986) provide an almost complete hierarchy covering each integer between 0 and $\mathbf{C}_\Omega(B_n)$. Again this will be presented in a style which permits the use of degenerate functions in deriving the hierarchy, although this and the theorem following it, may both be extended to permit only non-degenerate functions, with only a slightly more complex argument. The main result for network size assumes the basis B_2 , so that the dependence on Ω will be dropped from our notation. It will be clear from the proof that the same results hold for the basis $B_2 - \{\oplus, \iff\}$.

Theorem 2.12: (Paterson and Wegener, 1986) Let $n \geq 1$

$$c(r) = 1 \quad \text{for } 0 \leq r < \mathbf{C}(B_{n-1})$$

$$c(r) \leq n \quad \text{for } \mathbf{C}(B_{n-1}) \leq r < \mathbf{C}(B_n)$$

Proof: Let $f \in B_n$ with combinational complexity $\mathbf{C}(B_n)$. Suppose $\{\alpha_1, \dots, \alpha_r\}$ is the set of assignments to \mathbf{X}_n which satisfy f and that $\{p_1, \dots, p_r\}$ are the corresponding products of n literals. Consider the sequence of functions $\{f_0, \dots, f_r\}$ given by:

$$f_0 = 0 \quad ; \quad f_i = f_{i-1} \vee p_i \quad (0 < i \leq r)$$

(so $f_r = f$). Clearly we have in all cases:

$$\mathbf{C}(f_r) = \mathbf{C}(B_n) \quad (\text{By choice of } f)$$

$$\mathbf{C}(f_i) \leq \mathbf{C}(f_{i-1}) + n$$

which is sufficient to prove the second inequality of the theorem. To obtain the, optimal, first equality we decrease the gap in going from each f_{i-1} to f_i by interposing additional sequences of functions between them. So for this case let $f(\mathbf{X}_n - \{x_n\})$ be a function having combinational complexity $\mathbf{C}(B_{n-1})$ and f_0, f_1, \dots, f_r be the sequence of functions constructed, as before, from f . Modify this sequence by \vee -ing each component function with x_n . From the choice of sequence:

$$f_i \vee x_n = f_{i-1} \vee (y_1 y_2 \cdots y_{n-1}) \vee x_n$$

where each y_k is either x_k or its complement, depending on the product p_i . To compress the complexity gap between $x_n \vee f_i$ and $f_{i+1} \vee x_n$ substitute for each $f_i \vee x_n$ the sequence of $n-1$ functions given by:

$$f_i \vee x_n, f_i \vee (y_1 x_n), \dots, f_i \vee (y_1 y_2 \cdots y_j x_n)$$

(for $1 \leq j \leq n-2$), the precise literals used depending on the product p_{i+1} .

With this expanded sequence the first equality in the theorem statement follows. For each function can be computed from its predecessor by either replacing the input x_n by the appropriate literal y_{n-1} and then \vee -ing x_n with the output gate; or replacing the input x_n with $y_j \wedge x_n$. Both these transformations can be carried out using at most one additional gate. \square

For arbitrary complete bases from B_2 and for formula size slightly weaker results hold:

Theorem 2.13: (Paterson and Wegener, 1986) For any complete $\Omega \subseteq B_2$ there exists a constant k such that:

$$c_{\Omega}(r) \leq k \quad \text{for } 0 \leq r < \mathbf{C}_{\Omega}(B_{n-1})$$

$$c_{\Omega}(r) \leq kn \quad \text{for } \mathbf{C}_{\Omega}(B_{n-1}) \leq r < \mathbf{C}_{\Omega}(B_n)$$

$$l_{\Omega}(r) \leq kn \quad \text{for } 0 \leq r < \mathbf{L}_{\Omega}(B_n)$$

$$l(r) \leq n \quad \text{for } \Omega = B_2 \text{ and } 0 \leq r < \mathbf{L}(B_n)$$

Proof: Exactly as Thm(2.12) noting the definition of formulae and the result of Lemma(1.4). \square

Paterson and Wegener also present a nearly complete hierarchy for monotone network complexity with essentially the same argument. A very small improvement can be made for the upper range of this:

$$c_{\{\wedge, \vee\}}(r) \leq n \quad \text{for } \mathbf{C}^{\mathbf{m}}(M_{n-1}) \leq r < \mathbf{C}^{\mathbf{m}}(M_n)$$

as given in the paper. This improvement relies on a deep result from Korshunov (1981) and will be described in Chapter(3).

2.3) Relating Network Size and Network Depth

Theorem(2.4) presented 2 inequalities relating the combinational, formula and depth complexities of Boolean functions, namely:

$$\forall f(\mathbf{X}_n) \quad \mathbf{C}(f) \leq \mathbf{L}(f) < 2^{\mathbf{D}(f)}$$

We noted that these inequalities could not be improved. In this section inequalities in the reverse direction, which hold for all Boolean functions, are considered. The main result proved is the theorem from (Paterson & Valiant, 1976) which gives a lower bound on combinational complexity in terms of network depth having the form:

$$\forall f \in B_n \quad \mathbf{C}(f) \geq \frac{1}{4} \mathbf{D}(f) \log \mathbf{D}(f)$$

As a prelude to this we outline some results relating formula complexity and depth.

In these cases the method used is to construct a formula of "small" depth over some basis Ω_1 which is equivalent to a given formula over a basis Ω_2 by applying a result from (Brent, Kuck and Maruyama, 1973).

Lemma 2.4: (Brent et alia, 1973) Let F be an Ω -formula over \mathbf{X}_n and $|F|$ denote the total fanout from the inputs of F , which is equivalent to the total number of literals occurring in F using the sense of Defn(1.3).

$\forall m \ 1 < m \leq |F|$ there exists a subformula, $L\theta R$, of F ($\theta \in \Omega$) which satisfies:

- i) $|L\theta R| \geq m$
- ii) $|R| \leq |L|$ □

Observing that F is written as,

$$F(\mathbf{X}_n) = P(\mathbf{X}_n) \theta_1 (L\theta R)(\mathbf{X}_n) \theta_2 Q(\mathbf{X}_n)$$

where P and Q are Ω -formulae and $\theta_1, \theta_2 \in \Omega$, one may split F into two Ω -formulae; $(L\theta R)(\mathbf{X}_n)$ and $A(\mathbf{X}_n, y)$, the latter being $F(\mathbf{X}_n)$ with the subformula $L\theta R$ replaced by a new literal, y . Now suppose we are attempting to find a minimal depth equivalent formula over the basis B_2 , then using $A(\mathbf{X}_n, y)$ and $(L\theta R)(\mathbf{X}_n)$, a new equivalent formula is given by:

$$F'(\mathbf{X}_n) = A(\mathbf{X}_n, 1) \wedge (L\theta R)(\mathbf{X}_n) \vee (A(\mathbf{X}_n, 0) \Rightarrow (L\theta R))$$

With $d(k)$ to denote:

$$\max \{ \mathbf{D}(f) : f \text{ is realised by a formula of size } k - 1 \}$$

we have the relation:

$$d(|F|) \leq \max \{ d(|L\theta R|), d(|A|) \} + 2$$

(Some refinements are necessary for transforming into formulae over complete bases other than B_2 , however here we are only concerned with giving a brief description of the basic technique.)

The construction is applied recursively to the formulae $A(\mathbf{X}_n, 0)$, $A(\mathbf{X}_n, 1)$ and $L\theta R$ as necessary. Additionally a sequence $\langle r_i \rangle$ must be defined to specify the minimal value of $|L\theta R|$ at each stage, noting that $|A| = |F| - |L\theta R| \leq |F| - r_i$. The next theorem summarises some of the known relations between depth and formula size for specific complete bases. All of these are basically derived from the skeleton construction just reviewed.

Theorem 2.14: $\forall f \in B_n$

$$1) \quad \mathbf{D}_{\{\neg, \wedge\}}(f) \leq 2.88 \log \mathbf{L}(f) + O(1)$$

$$2) \quad \mathbf{D}(f) \leq 2.465 \log \mathbf{L}(f) + O(1)$$

$$3) \mathbf{D}_{U_2}(f) \leq 1.81 \log \mathbf{L}_{U_2}(f) + O(1)$$

U_2 being the basis $B_2 - \{\oplus, \iff\}$

Proof: (1) is from McColl (1977), (2) from Spira (1971a) and (3) is due to Preparata and Muller (1976). \square

(3) improves the earlier result of (Barak and Shamir, 1976). \square

The relation between size and depth is obtained in a similar style by constructing a small depth network equivalent to one of minimal complexity. In what follows the basis B_2 is assumed. For a given combinational network, T , $e(T)$ will denote the total number of wires leaving *gate* nodes in T . For $v, w \in \mathbf{N}$, $B(v)$ and $A(w)$ are defined respectively by:

$$\max \{ \mathbf{D}(f) : f \text{ is realised by a network } T \text{ having } e(T) \leq v \}$$

$$\max \{ v : B(v) \leq w \}$$

For any $v > 0$, $B(v) \leq 1 + B(v - 1)$. To see this, consider any network T with $e(T) = v$. If any gate of T , which has only variables as inputs, is replaced by a new variable, then the resulting network T_1 computes a new function, has $e(T_1) \leq v - 1$ and so depth no more than $B(v - 1)$. Thus $B(v) \leq 1 + B(v - 1)$ because the original function has depth at most one greater.

Theorem 2.15: (Paterson & Valiant, 1976) $\forall f \in B_n$

$$\mathbf{C}(f) \geq \frac{1}{4} \mathbf{D}(f) \log \mathbf{D}(f) - o(\mathbf{D}(f))$$

Proof: Let T be an optimal network realising some function f at a gate t and having $e(T) = v > 0$. Suppose the gates of T are partitioned into 2 sets, \mathbf{Y} and \mathbf{Z} say, with no gate of \mathbf{Z} preceding a gate of \mathbf{Y} . t must be in \mathbf{Z} if \mathbf{Z} is non-empty. Let M be the set of those gates in \mathbf{Y}

which supply an input to some gate in \mathbf{Z} and let $m = |M|$. We wish to consider the gates of \mathbf{Y} as forming a network computing some multiple output function of \mathbf{X}_n , and the gates of \mathbf{Z} as a network computing a Boolean function of \mathbf{X}_n and $\{res(g) : g \in M\}$. We use Y to denote the network comprising the gates in \mathbf{Y} , the inputs \mathbf{X}_n and with wires as in T . Similarly Z denotes the network having inputs \mathbf{X}_n together with the outputs of M , gates from \mathbf{Z} and wires as in T . Further let $e(Y) = y$, $e(Z) = z$. Since each gate in M supplies at least one wire between \mathbf{Y} and \mathbf{Z} , it is immediate that:

$$y + z + m \leq v \quad (2.11)$$

Suppose a single gate is moved from \mathbf{Y} into \mathbf{Z} . This decreases y by at most 2 and m by at most 1, so the partition of T into \mathbf{Y} and \mathbf{Z} can be chosen to satisfy:

$$|2y + m - v| \leq 2 \quad (2.12)$$

Let $u = \max\{y, z\}$. From (2.11) we have $2u + m \leq v + |y - z|$ and $z \leq v - y - m$. From (2.12) $|y - z| \leq |2y + m - v| \leq 2$. Thus:

$$2u + m \leq v + 2 \quad (2.13)$$

Now Y is a network realising each of the functions in $\{res(g) : g \in M\}$, so all of these can be computed in depth $B(y)$. By using this set of functions as input to a minimal depth network equivalent to Z we obtain a network realising f showing that:

$$\mathbf{D}(f) \leq B(y) + B(z) \leq 2B(u) \quad (2.14)$$

Consider the function $f'(\mathbf{X}_n, s_1, \dots, s_m)$ in B_{n+m} computed by Z at t . From the choice of Z it is clear that:

$$f = f'(\mathbf{X}_n, res(g_1), \dots, res(g_m)) \quad (2.15)$$

where $\langle g_1, \dots, g_m \rangle$ is M under some ordering.

Obviously $\forall \alpha \in \{0, 1\}^m$

$$\mathbf{D}(f'(\mathbf{X}_n, \alpha)) \leq \mathbf{D}(f'(\mathbf{X}_n, s_1, \dots, s_m)) \leq B(z)$$

Recall that the disjunctive expansion of f' about $\langle s_1, \dots, s_m \rangle$ is given by:

$$\bigvee_{\alpha \in \{0, 1\}^m} \delta_\alpha(s_1, \dots, s_m) \wedge f'(\mathbf{X}_n, \alpha)$$

With this a new network realising $f(\mathbf{X}_n)$ can be constructed as follows. For each $\alpha \in \{0, 1\}^m$, Z_α is the network formed from Z by replacing each input $g_i \in M$ by the corresponding constant a_i of α and absorbing these into their successor gates (cf. Lemma(1.3)). Z_α computes $f'(\mathbf{X}_n, \alpha)$ at t . Any assignment, β , to \mathbf{X}_n the inputs of Y induces some m -tuple at the gates in M . For any $\alpha \in \{0, 1\}^m$ let $\chi_\alpha(\mathbf{X}_n)$ be the Boolean function which is 1 if and only if this m -tuple corresponds to α . Noting our identity (2.15) for f and the earlier disjunctive expansion of f' we have:

$$f(\mathbf{X}_n) = \bigvee_{\alpha \in \{0, 1\}^m} \chi_\alpha(\mathbf{X}_n) \wedge f'(\mathbf{X}_n, \alpha) \quad (2.16)$$

Since every $\chi_\alpha(\mathbf{X}_n)$ is just a product of the, possibly negated, outputs of the gates in M , so $\chi_\alpha(\mathbf{X}_n)$ can always be realised in depth at most $\lceil \log m \rceil + B(y)$. Therefore the network for f built according to (2.16) has depth not exceeding:

$$\max \{B(z), B(y) + \lceil \log m \rceil\} + m + 1$$

Hence:

$$\mathbf{D}(f) \leq \max \{B(z), B(y) + \lceil \log m \rceil\} + m + 1$$

$$\leq B(u) + \lceil \log m \rceil + m + 1$$

$$\leq B(u) - 2u + v + 3 + \lceil \log m \rceil \quad \text{via (2.13)} \quad (2.17)$$

So for a single fixed partition of T we have two inequalities, (2.14) and (2.17), giving an upper bound on $\mathbf{D}(f)$. Let us fix f and T for some chosen r in such a way that $e(T) = v = A(r) + 1$ and $\mathbf{D}(f) > r$. From (2.14)

$$B(u) > \lfloor r/2 \rfloor \quad \text{or equivalently} \quad u > A(\lfloor r/2 \rfloor) \quad (2.18)$$

The right-hand side of (2.17) is easily maximised by choosing $u = A(\lfloor r/2 \rfloor) + 1$, $m = v + 2 - 2u$ since this expression is a decreasing function of u (recall $B(u) \leq B(u-1) + 1$) and an increasing function of m . From this:

$$r < \mathbf{D}(f) \leq \lfloor r/2 \rfloor + 1 - 2(A(\lfloor r/2 \rfloor) + 1) + v + 3 + \lfloor \log v \rfloor$$

which simplifies to

$$v + \lfloor \log v \rfloor > 2A(\lfloor r/2 \rfloor) + \lceil r/2 \rceil - 2 \quad (2.19)$$

T was fixed so that $e(T) = v = A(r) + 1$ so (2.19) yields a recurrence inequality for $A(r)$ for any r . To solve this let:

$$H(r) = \frac{1}{2} r \log r + 2 \log r - kr$$

then

$$2H(\lfloor r/2 \rfloor) + \lceil r/2 \rceil - 2 > H(r) + 1 + \lceil \log r \rceil$$

for all $k \geq 0$ and sufficiently large r . An easy induction on r can be used to prove that for some k

$$A(r) \geq H(r) \geq \frac{1}{2} r \log r - O(r)$$

Finally since for any network T , $e(T) \leq 2\mathbf{C}(T)$ we have shown:

$$C(f) \geq \frac{1}{4} D(f) \log D(f) - O(D(f)) \quad \square$$

2.4) Lower Bounds on Specific Boolean Functions

The preceding pages have largely been concerned with properties that either all or almost all Boolean functions possess. Focusing now on specific families, it would be appropriate to present as coda to Shannon's theorem, a proof that some explicitly defined^{b)} family of functions had combinational complexity $\Omega(2^n/n)$. At present no such proof exists. However, deriving bounds of this magnitude is perhaps rather too ambitious and one might be content, for subsequent development, with simply exponential or polynomial or even just superlinear complexity, knowing from the various hierarchy theorems, that families of this difficulty abound. Again, despite some considerable effort encompassing almost 40 years, no such results are known. Establishing results of this nature remains one of the most fundamental objectives for complexity theory, the lack of progress to date serving to highlight one frustrating and challenging aspect of the problem; the apparent paradox that much is known about the complexity of functions in general, but little about the difficulty of particular cases.

So the most powerful techniques currently available yield only linear lower bounds on combinational complexity. Here are presented three progressively stronger results; a theorem of Schnorr (1974) which gives lower bounds of $2n - 3$ on an important subset of B_n ; the method of Stockmeyer (1977) which allows bounds of $2.5n - 5$ to be obtained for certain symmetric functions; and finally the $3n$ lower bound of Blum (1984a) which is the best achieved to date. It should

b) Formally, we shall regard a family $[f_n]$ as "explicitly defined" if there is a *TM* which given n in unary as input, outputs the truth-table for $f(\mathbf{X}_n)$, in time polynomial in 2^n .

be noted that the concepts employed by both Stockmeyer and Blum rely heavily on an approach developed in the earlier $2.5n$ lower bound of Paul (1977), which will not be presented explicitly.

That the extant theory is devoid of all, save existential, expositions that specific functions are difficult to realise, may be attributed to the impotence of what is essentially still the only generally applicable paradigm for reasoning about combinational complexity: inductive gate elimination. The form this takes is simply described. Consider any family, $[f_n]$ of Boolean functions. Suppose $s(n)$ is a function from $\mathbf{N} \rightarrow \mathbf{N}$ and it is desired to prove a lower bound of $s(n)$ on the combinational complexity of $[f_n]$. This may be accomplished by showing that $\mathbf{C}(f_c) \geq s(c)$, as the inductive base, c denoting the number of arguments to the smallest instance in the family. The inductive step assumes that $\mathbf{C}(f_i) \geq s(i)$, $\forall c \leq i < n$ and consists of some analysis which proves:

\forall optimal combinational networks, T , realising f_n , there exists a partial assignment π such that $f_n^{|\pi} = f_{n-|\pi|}$ (i.e a lower indexed member of the family). Furthermore applying π to T and simplifying, using Lemma(1.3), eliminates at least $k(n)$ gates. Now since

$$\mathbf{C}(f_n) \geq \mathbf{C}(f_{n-|\pi|}) + k(n) \geq s(n - |\pi|) + k(n)$$

it follows from the inductive hypothesis that if $k(n)$ is large enough then $\mathbf{C}(f_n) \geq s(n)$.

The inductive base should be relatively easy. The limitations of the method become apparent in proving the inductive step, where two difficulties arise; the requirement to project onto a smaller instance in the family; and the need to eliminate sufficient gates in order to make the inductive process succeed. The first of these may often be circumvented by using a broader concept of family. In this way instead

of attempting to prove bounds for a single function one aims to derive results for the complexity of all families which possess some property, e.g symmetric, threshold etc. This need not complicate the inductive base and can permit considerable latitude in the choice of partial assignment, since now it is sufficient to project onto any smaller function which has the required property.

It is the second constraint which presents the major obstacle to more substantial results. To encapsulate all optimal combinational networks for f_n in the inductive step, the normal mechanism is to proceed by an exhaustive case analysis examining the fanout of the input nodes. The purely inductive argument outlined above is sufficient to derive Schnorr's $2n$ results. However, for proving larger bounds, often it happens that there is some case where not enough gates can be eliminated at once. To deal with this it is necessary to rely on the knowledge gleaned from the other cases about the structure of optimal networks for which induction fails and thereby prove that such networks contain the requisite number of gates directly, e.g Blum's analysis effectively reduces to examining a network in which all inputs have fanout 1 and enter \wedge -type gates, all other cases being handled inductively. Such proofs are notable for the great complexity and considerable technical sophistication of the arguments used to handle the final cases. Paul (1977) was the first to develop these with a method asserting the existence of gates in the final network, which were potentially quite distant from the inputs. The techniques of that paper have since been modified by Stockmeyer and enhanced by Blum. The reader should be aware that the earlier $3n$ lower bound "proof" of Schnorr (1980) is now known to be incomplete, cf Blum (1984a).

Below we shall frequently make use of Lemma(1.3) without directly referring to it.

The results of Schnorr (1974) and Stockmeyer (1977) pertain to certain members of S_n , the class of Boolean symmetric functions, Stockmeyer's bounds being a development of Schnorr's.

For $f \in S_n$, $\mathbf{w}(f) = w_0 w_1 \cdots w_n \in \{0, 1\}^{n+1}$ denotes the spectrum of f . SW is the set of strings in $\{0, 1\}^4$ which contain 3 distinct substrings of length 2, i.e

$$SW = \{ 0100, 0010, 0110, 0011, 1011, 1101, 1001, 1100 \}$$

For $k \geq 0$, $n \geq 2k + 3$ we consider the subset $F_{n,k}$ of S_n which consists of:

$$\{ f \in S_n : \mathbf{w}(f) \in \{0, 1\}^{\geq k} \cdot SW \cdot \{0, 1\}^{\geq k} \}$$

(Unless otherwise stated, it will be assumed throughout that $k \geq 0$.)

A lower bound on the complexity of functions in $F_{n,k}$ follows directly from any lower bound on

$$\mathbf{C} = \min \{ \mathbf{C}(f) : f \in F_{n,k} \}$$

The results presented show that:

$$\mathbf{C} \geq 2n - 3 \quad \forall n \geq 3 \quad (\text{Schnorr, 1974})$$

$$\mathbf{C} \geq 2n + k - 3 \quad \forall n \geq 2k + 3 \quad (\text{Stockmeyer, 1977})$$

All but 8 functions in S_n are in $F_{n,0}$, these being the constant functions, and six functions having complexity $n - 1$.

To further simplify the description we employ the following notation of Stockmeyer.

For $n \geq 3$, let $f \in S_n$ with $\mathbf{w}(f) = w_0 w_1 \mathbf{u} w_{n-1} w_n$, where $\mathbf{u} \in \{0, 1\}^{n-3}$. Functions f_{00} , f_{01} , f_{10} , f_{11} in S_{n-2} are given by spectra:

$$\begin{aligned}\mathbf{w}(f_{00}) &= w_0 w_1 \mathbf{u} \\ \mathbf{w}(f_{01}) &= \mathbf{w}(f_{10}) = w_1 \mathbf{u} w_{n-1} \\ \mathbf{w}(f_{11}) &= \mathbf{u} w_{n-1} w_n\end{aligned}$$

(i.e the spectra resulting by setting two inputs to 0, 0 and 1 or 1 respectively.)

Functions f_0, f_1 in S_{n-1} are defined similarly from f so that:

$$\mathbf{w}(f_0) = w_0 w_1 \mathbf{u} w_{n-1} \quad ; \quad \mathbf{w}(f_1) = w_1 \mathbf{u} w_{n-1} w_n$$

For arbitrary c, d in $\{0, 1\}$ and f in S_n the functions f_{cd} in S_{n-2} and f_c in S_{n-1} are given in the obvious way using the preceding definitions.

From these we have:

Lemma 2.5:

- (I) If $f \in F_{n,k}$ then f_{00}, f_{01}, f_{11} are all distinct functions.
- (II) If $f \in F_{n,k}$, where $k \geq 1$ then none of $f_{00}, f_{01}, f_{11}, f_0$ or f_1 are constant functions.

Proof: The lemma may be verified directly from the definition of $F_{n,k}$. \square

The next result is an important lemma, due to Schnorr, which shows that in any optimal network realising some $f \in F_{n,k}$ we can identify an input having fan-out at least 2.

Lemma 2.6: (Schnorr, 1974) Let $f \in F_{n,k}$ and T be an optimal network computing f at some node t . There exists some x_i in \mathbf{X}_n such that $\phi(x_i) \geq 2$ in T .

Proof: Let g be a gate in T whose distance from the output gate is maximal. Both inputs of g must be (distinct) inputs of T otherwise it would be possible to select a gate at a greater distance from t .

Without loss of generality, let x_i, x_j be the inputs of g and suppose that $\phi(x_i) = \phi(x_j) = 1$. Since at least 2 of $res(g)(0,0)$, $res(g)(0,1)$, $res(g)(1,1)$ must be identical and since T depends on x_i, x_j only via g it follows that two of:

$$f_{00}(\mathbf{X}_n - \{x_i, x_j\}), f_{01}(\mathbf{X}_n - \{x_i, x_j\}), f_{11}(\mathbf{X}_n - \{x_i, x_j\})$$

are identical. But then from Lemma(2.4) f cannot be in $F_{n,k}$ and this contradiction proves the lemma. \square

Now we can prove our first lower bound using the inductive gate elimination method.

Theorem 2.16: (Schnorr, 1974) $\forall n \geq 3, \mathbf{C} \geq 2n - 3$

Proof: It is convenient to choose $n = 2$ as the base of the induction, defining: \mathbf{C} as

$$\min \{ \mathbf{C}(f) : f \in S_2 \text{ and } f \text{ is not a constant function} \}$$

(The alternative would be to show directly that all functions in $F_{3,0}$ require at least 3 gates; since $|F_{3,0}| = |SW| = 10$ this latter would be somewhat tedious.)

Base: $n = 2$ Obvious

Inductive Step: Assume the result holds for all values $2 \leq r < n$. To show the theorem holds for $F_{n,0}$ consider any $f \in F_{n,0}$ and an optimal network, T , realising f . Let x_i , with $\phi(x_i) \geq 2$, be the input identified in the proof of Lemma(2.6). Since $f \in F_{n,0}$, there is some c in $\{0, 1\}$, such that $f_c \in F_{n-1,0}$. Set $x_i = c$, simplify T and rename the remaining inputs x_1, \dots, x_{n-1} . The new network computes f_c and contains 2 fewer gates because $\phi(x_i) \geq 2$. Thus:

$$\mathbf{C}(f) \geq \mathbf{C}(f_c) + 2 \geq \mathbf{C} + 2 \geq 2(n-1) - 3 + 2 = 2n - 3$$

from the Inductive hypothesis. \square

To obtain larger bounds it is necessary to examine the structure of optimal networks in more detail. The lower bound on \mathbf{C} is a consequence of the following lemma.

Lemma 2.7: (Stockmeyer, 1977) $\forall k \geq 1, \forall n \geq 2k + 3$

$$\mathbf{C} \geq \min \{ \mathbf{C} + 3, \mathbf{C} + 5 \}$$

Proof: Let $f \in F_{n,k}$ and T be an optimal combinational network realising f at t . It is assumed that of all such networks T is chosen so that the quantity $X - out$, being the total fanout from input nodes, is minimal. This assumption is required only once at the conclusion of the proof. It will be shown that at least one of the following holds:

- R1) There exists an input x_i of T such that setting $x_i = c \in \{0, 1\}$ eliminates at least 3 gates.
- R2) There exist distinct inputs x_i, x_j of T such that setting $x_i = 0$ and $x_j = 1$ eliminates at least 5 gates. Since

$$f^{|x_i=c} = f_c \in F_{n-1,k-1} \quad \text{and} \quad f^{|x_i=0, x_j=1} = f_{01} \in F_{n-2,k-1}$$

(after renaming the inputs as appropriate) this proves the lemma. To establish that (R1) or (R2) holds consider any optimal network realising f . We proceed by a case analysis.

Case 1: There exists an input x_i of T such that $\phi(x_i) \geq 3$

The successors of x_i must be distinct and so setting $x_i = 0$ eliminates 3 gates.

Case 2: There exists an input x_i of T such that $\phi(x_i) = 2$ and x_i enters an \wedge -type gate.

Let g_1, g_2 be the successors of x_i and without loss of generality assume that g_1 is the \wedge -type gate, the other input of which is some gate h . In this case:

$$res(g_1) = ((x_i)^a \wedge res(h)^b)^c$$

Setting $x_i = \bar{a}$ eliminates g_1 and g_2 . In addition, since $res(g_1)$ is a constant function under this assignment, all the successors of g_1 can also be eliminated. For from Lemma(2.5) $g_1 \neq t$; from optimality some successor of g_1 must differ from g_2 , otherwise g_1, g_2 could be replaced by a single gate. So 3 gates in total may be removed.

We are left with just one case. For this x_i, x_j and d will denote the nodes of T identified in the proof of Lemma(2.6)

Case 3: Cases(1) and (2) do not hold and $\phi(x_j) = 2$

The case $\phi(x_j) = 1$ is almost identical and so will not merit separate consideration. The mechanics of the proof are not affected by this detail. x_i must enter 2 \oplus -type gates, d and g_1 say. x_j enters d and some gate h_1 , which is also \oplus -type. In this case we must resort to examining gates which are deeper in the network; an argument based on the methods of Paul (1977) is employed.

Let $\langle g_1, g_2, \dots, g_p \rangle$ be a path in T with the properties:

- i) $\forall 1 \leq i \leq p$ g_i is an \oplus -type gate.
- ii) $\forall 1 \leq i < p$ $\phi(g_i) = 1$
- iii) $\phi(g_p) > 1$ or g_p enters an \wedge -type gate or $g_p = t$ the output of T .

A path $\langle h_1, h_2, \dots, h_q \rangle$ is defined analogously. It should be obvious that both paths exist. For the path from h_1, s_i will denote the node which supplies the other (than x_j or h_{i-1}) input of h_i . The sub-network identified is depicted in Figure(2.2).

This network has several important properties.

Property 1: $\forall 1 \leq k \leq p, \forall 1 \leq l \leq q$ $g_k \neq h_l$

Figure 2.2

Proof: If $g_k = h_l$ then

$$res(g_k) = x_i \oplus x_j \oplus r(\mathbf{X}_n)$$

for some function $r \in B_n$. t depends on x_i, x_j only via d and g_k but:

$$\begin{aligned} res(d) \Big|_{x_i=x_j=0} &= res(d) \Big|_{x_i=x_j=1} \\ res(g_k) \Big|_{x_i=x_j=0} &= res(g_k) \Big|_{x_i=x_j=1} \end{aligned}$$

hence $f^{|x_i=x_j=0} = f^{|x_i=x_j=1}$ and this contradicts $f \in F_{n,k}$. \square

Without loss of generality it may be assumed that T does not contain any path from g_p to h_q ; for since T is acyclic there cannot be both a path from g_p to h_q and a path from h_q to g_p .

Property 2: $\forall 1 \leq k \leq p, \forall 1 \leq l \leq q$ $g_k \neq s_l$ and there is no path from g_k to s_l .

Proof: Suppose that $g_k = s_l$. If $k < p$ then $g_{k+1} = h_l$ and this contradicts Property(1). If $k = p$ then there is a path from g_p to h_q contradicting our previous assumption. Since $g_k \neq s_l$ any path from g_k to s_l must be via g_p and this again would yield a path from g_p to h_q . \square

We now make a simple modification to the network of Fig(2.2).

If $q > 1$ then

Delete the wires $\langle x_j, h_1 \rangle, \langle s_q, h_q \rangle$

Add wires $\langle x_j, h_q \rangle, \langle s_q, h_1 \rangle$

See Figure(2.3) in which h_q has been renamed h and s is the gate supplying the other input of h .

Property 3:

- i) $g_1 \neq s$ and there is no path from g_1 to s
- ii) $\phi(h) \geq 2$ or h enters an \wedge -type gate or $h = t$
- iii) $d \neq s$

Proof: (i) follows directly from Property(2) and (ii) is immediate from the choice of h_q . For (iii) if $d = s$ then $res(h) = (x_i)^a$ for some $a \in \{0, 1\}$ which would contradict the optimality of T . \square

Property(3) guarantees that the six nodes in Figure(2.3) are different.

Figure 2.3

The remainder of the proof concentrates on this sub-network and its environment. The objective is to establish two further properties:

- i) $|Elim| \geq 2$, where $Elim$ is the set of gates which have an input from d or h .
- ii) T may be rewired to a network T' realising f_{01} and in which the nodes x_i, x_j, d and h all compute constant functions.

For $e \in \{0, 1\}$, T_e is the network obtained from T by changing the sub-network of Fig(2.3) as follows:

- E1) Delete the nodes x_i, x_j and all wires leaving them from T .
- E2) Replace the wire $\langle s, h \rangle$ by a wire $\langle s, g_1 \rangle$.
- E3) Replace d by the constant function, b , associated with $res(d)$, i.e $res(d) = b \oplus x_i \oplus x_j$.

E4) Replace h by the constant function $e \oplus c$ where $res(h) = c \oplus x_j \oplus res(s)$ in T .

The net effect of (E1)-(E4) is that of replacing x_i by the function $[res(s)]^{\bar{e}}$ and x_j by the function $[res(s)]^e$. Clearly T_e realises f_{01} for both $e = 0$ and $e = 1$. T_e contains two fewer gates (d, h) than T and two nodes computing constant functions (from (E3) and (E4)). We now show that at least 3 gates in addition to d and h may be eliminated from T in forming T_e .

Property 4: Let $e \in \{0, 1\}$.

i) For any gate $r \neq s$ in T_e

$$\{u : r \text{ enters } u \text{ in } T_e\} = \{u : r \text{ enters } u \text{ in } T\}$$

ii) For any node $y \neq s$ in T_e if y computes a constant function then $\phi(y) \geq 1$.

iii) $\phi(d) \geq 1, \phi(h) \geq 1$ in T .

Proof: (i) is immediate from the definition of T_e ; (ii) follows from (i) and the fact that f_{01} is not a constant function; (iii) is consequence of (ii) and Property(3)(ii). \square

(If $\phi(x_j) = 1$ then T_e is formed as before but the wire $\langle s, g_1 \rangle$ is not added. Property(4) still holds as stated)

We can now resume the case analysis, this time centred on the size of $Elim$.

Case 3.1: $|Elim| \geq 3$

Choose $e = 0$, then the gates d, h and at least 3 gates in $Elim$ can be deleted from T in creating T_e . Since T is optimal $d, h \notin Elim$.

Case 3.2: $|Elim| = 2$

Let $Elim = \{u, v\}$; u being a successor of d and v of h , where $\{u, v, d, h\}$ are distinct by virtue of Prop(3)(ii) and (4)(iii)

Case 3.2.1: $\phi(h) = 2$ (See Figure(2.4))

Figure 2.4

Consider the network T_0 . In this $res(u)$ is a constant function, since h and d become constants, thus from Prop(4)(ii) $\phi(u) \geq 1$. If v is a successor of u then $res(v)$ in T_0 is also constant and $\phi(v) \geq 1$. So here d , h , u , v and all successors of v can be eliminated from T in forming

T_0 . Alternatively if u does not enter v , then all successors w of u together with d, h, u and v can be eliminated.

Case 3.2.2: $\phi(h) = 1$

From Prop(3)(iii) v must be an \wedge -type gate, so we can choose some e , depending on $op(v)$, so that $res(v)$ is constant in T_e . If v enters any gate $w \neq u$ then the 5 gates d, h, u, v and w may be deleted. If v enters only u then, for e chosen as before, $res(u)$ is constant in T_e and so has some successor $w \notin \{d, h, v, u\}$, and again 5 gates may be eliminated.

It remains to dispose of the case where $Elim$ contains but a single gate. The final property establishes the impossibility of this occurring and uses the assumption on the minimality of $X - out$ made at the start of this proof.

Property 5: $|Elim| \neq 1$

Proof:

Suppose the contrary, so that we have the sub-network of Figure(2.5). The gate u must be \wedge -type. If s is not an input of T then re-wire T as depicted in Figure(2.6), to a new network T' , amending the gate operations of d, h and u to ensure that $res(u)$ in T is the same as $res(u')$ in T' . That this is always possible follows from the identity:

$$(x \iff y) \wedge (y \iff z) = (x \iff z) \wedge (y \iff z)$$

Now T' computes f using no more gates but the total input fanout of T' is less than that of T , contradicting the initial choice of T .

On the other hand, suppose s is an input of T , x_l say. From optimality $l \neq i$ and $l \neq j$. In this case, since u is \wedge -type, we can fix x_i

Figure 2.5

and x_l to constants so that $res(u)$ is constant. The resulting network is independent of x_j but should compute a non-constant symmetric function of $n - 2$ arguments. This contradiction establishes Property(5) and completes the proof of Lemma(2.7). \square

It is now easy to prove:

Theorem 2.17: $\forall k \geq 0, \forall n \geq 2k + 3$

$$\mathbf{C} \geq 2n + k - 3$$

Figure 2.6

Proof: By induction on k . The inductive base, $k = 0$, is just Theorem(2.16). Assuming the result holds for all values less than k and applying Lemma(2.7) we obtain from the Inductive hypothesis:

$$\mathbf{C} \geq \min \{2(n-1) + (k-1) - 3 + 3, 2(n-2) + (k-1) - 3 + 5\}$$

and this quantity is $2n + k - 3$. \square

Special cases of interest are the threshold and congruence functions.

Corollary 2.4: $\forall k \ 2 \leq k \leq n-1$

$$\mathbf{C}(T_k^n) \geq 2n + \min \{k-2, n-k-1\} - 3$$

$\forall k \ 3 \leq k \leq n-1$

$$C(C_k^n) \geq 2.5n - k/2 - 4$$

Proof: The first inequality follows from Thm(2.17) since:

$$w(T_k^n) = 0^{k-2} \cdot 0011 \cdot 1^{n-k-1}$$

The second inequality is obtained by fixing $p = (n-k-1)/2$ and observing that:

$$w(C_k^n) \in \{0, 1\}^{\geq p} \cdot \{0100, 0010\} \cdot \{0, 1\}^{\geq p}$$

□

In particular we have:

$$C(T_{n/2}^n) \geq 2.5n - 5$$

$$C(C_4^n) \geq 2.5n - 6$$

As Stockmeyer demonstrates this last is, to within an additive constant, the best possible, his paper proving that:

$$C(C_4^n) \leq 2.5n$$

By fixing $k = \lceil (n-3)/2 \rceil$, $m = \lfloor (n-3)/2 \rfloor$ it is easy to see that the number of spectra of the form $\{0, 1\}^k \cdot SW \cdot \{0, 1\}^m$ is $2^3 \cdot 2^k \cdot 2^m = 2^n$ and hence at least half of the functions in S_n have combinational complexity $\geq 2.5n - 5$.

Stockmeyer's argument is atypical in that the results are effected entirely by an inductive process. This is not so in Paul's original 2.5n lower bound where the structure of the function examined necessitates the investigation of further cases. For completeness we state Paul's

result below.

Theorem 2.18: (Paul, 1977) Let $\alpha_1 = \langle a_1, a_2, \dots, a_m \rangle$, $\alpha_2 = \langle a_{m+1}, \dots, a_{2m} \rangle$ where $m = \lceil \log n \rceil$. For a given assignment from $\{0, 1\}^m$, α_i encodes, in binary, some integer between 0 and $2^m - 1$ in the obvious way. (α_i) denotes this value.

$$f(\alpha_1, \alpha_2, q, \mathbf{X}_n) : \{0, 1\}^{n+2m+1} \rightarrow \{0, 1\}$$

is given by:

$$(q \wedge x_{(\alpha_1)} \wedge x_{(\alpha_2)}) \vee (\bar{q} \wedge (x_{(\alpha_1)} \oplus x_{(\alpha_2)}))$$

(Note: f is arbitrary if $(\alpha_i) > n$ or $(\alpha_i) = 0$)

$$\mathbf{C}(f) \geq 2.5n - 2 \quad \square$$

Two strands run through the proof of this bound; the use of the "indirect address" fields; and the application of the "control" bit, q . Both are important in gaining information about the form of optimal networks realising f .

The $3n$ lower bound of Blum (1984a), which is presented next, also exploits these concepts. The function considered is a little more complex than Paul's, involving an additional address field $\alpha_3 = \langle a_{2m+1}, \dots, a_{3m} \rangle$.

$f(\alpha_1, \alpha_2, \alpha_3, q, \mathbf{X}_n)$ is defined to be:

$$x_{(\alpha_1)}^q \wedge (x_{(\alpha_2)} \oplus x_{(\alpha_3)})$$

Since Paul's function is a special case of this it follows from Thm(2.18) that $\mathbf{C}(f) \geq 2.5n - 2$.

Theorem 2.19: (Blum, 1984a) For $f(\alpha_1, \alpha_2, \alpha_3, q, \mathbf{X}_n)$ as given above:

$$\mathbf{C}(f) \geq 3n - 3$$

Proof: For any s , $1 \leq s \leq n$, let $P(s)$ be the assertion:

$C(f) \geq 3s - 3$ for all $f: \{0, 1\}^{n+3m+1} \rightarrow \{0, 1\}$ such that there exists $S \subseteq \{1, 2, \dots, n\}$ of cardinality s , satisfying:

$$\forall \langle (\alpha_1), (\alpha_2), (\alpha_3) \rangle \in S^3$$

$$f(\alpha_1, \alpha_2, \alpha_3, q, \mathbf{X}_n) = x_{(\alpha_1)}^q \wedge (x_{(\alpha_2)} \oplus x_{(\alpha_3)})$$

We prove the veracity of $P(s)$ by induction on s , thus again the lower bounds are established for a class of functions having a particular property, albeit a rather specialised one.

Inductive Base: $s = 1$, $s = 2$

The case $s = 1$ is obvious. The case $s = 2$ follows by observing that any function with the property of interest having $s = 2$, depends essentially on at least 4 inputs (2 data inputs, at least 1 address bit and 1 control bit) and thus requires at least 3 gates to be computed.^{c)}

Inductive Step: Let $P(s')$ hold for all values $s' < s$. Consider any f satisfying the property stated above for some set S of cardinality s . Let T be an optimal combinational network realising f at t . We proceed by a case analysis which examines the environment of x_i such that $i \in S$. The cases:

Case 1: $\exists i \in S$ such that $\phi(x_i) \geq 3$

Case 2: $\exists i \in S$ such that $\phi(x_i) = 2$ and x_i enters an \wedge -type gate.

are similar to the corresponding cases in Lemma(2.7) and are left to the reader to confirm.

c) Extending the inductive base to cover $s = 2$ is a technical convenience; at certain points in the proof of the inductive hypothesis it will be necessary to select 3 different elements of S .

Case 3: $\exists i \in S$ such that $\phi(x_i) \leq 2$ and x_i enters only \oplus -type gates.

Case(3) of Lemma(2.7) established the existence of a path $\langle g_1, \dots, g_p \rangle$ of \oplus -type gates satisfying:

- i) x_i enters g_1
- ii) g_k enters g_{k+1} $1 \leq k < p$
- iii) $\phi(g_k) = 1$ $1 \leq k < p$
- iv) $\phi(g_p) \geq 2$ or g_p enters an \wedge -type gate
- v) If r_k denotes the node supplying the other input of g_k then $res(r_k)$ does not depend on x_i , $\forall 1 \leq k \leq p$.

(v) is just a reformulation of the properties present by assuming the absence of certain paths in T .

Clearly:

$$res(g_p) = [x_i \oplus \bigoplus_{j=1}^p res(r_j)] \oplus e \quad (e \in \{0, 1\})$$

$$= x_i \oplus h(\mathbf{X}_n - \{x_i\}), \text{ say}$$

So replacing x_i by the function h or \bar{h} renders $res(g_p)$ a constant function. With this we can proceed to eliminate 3 gates from T . Consider the two possibilities for g_p .

A) $\phi(g_p) \geq 2$

Compute h from $res(r_1), \dots, res(r_p)$ using at most $p - 1$ \oplus -type gates and replace x_i by h . $res(g_p)$ is now constant and so all gates $\{g_1, \dots, g_p\}$ and the successors of g_p may be eliminated. The new network contains 3 fewer gates and satisfies $P(s - 1)$, for the set $S - \{i\}$ by the inductive hypothesis.

B) $\phi(g_p) = 1$

Therefore g_p enters an \wedge -type gate, u say. An easy argument, given in Paul (1977) establishes $\phi(u) \geq 1$ (i.e $u \neq t$). As in (A) replace x_i by h or \bar{h} to ensure that $res(u)$ is constant. This allows $\{g_1, \dots, g_p\}$, u and all successors of u to be eliminated. Again this leaves a network containing 3 fewer gates than T and satisfying $P(s-1)$ for the set $S - \{i\}$.

So it may be assumed that none of Cases(1-3) are true for T . Thus we have $\forall i \in S$:

$\phi(x_i) = 1$ and x_i enters an \wedge -type gate C_i , say.

Some further properties of T , in this case, are now proved.

Property 1: $\forall i, j \in S, (i \neq j) \ C_i \neq C_j$

Proof: Suppose the contrary. Then there exists some $e \in \{0, 1\}$ for which $res(C_i)^{|x_i=e}$ is constant, hence replacing x_i by e renders T independent of x_j . But $f^{|x_i=e}$ still depends on x_j , e.g fix $(\alpha_1) = (\alpha_2) = i, (\alpha_3) = j, q = e$ and the remaining variables arbitrarily. In this case $f = e \oplus x_j$. \square

Case 4: $\exists i \in S$ such that $\phi(C_i) \geq 2$

Choose $e \in \{0, 1\}$ for which $res(C_i)^{|x_i=e}$ is a constant function. Then fixing $x_i = e$ allows C_i and its ≥ 2 successors to be eliminated.

We now have only:

Case 5: $\forall i \in S \ \phi(C_i) = 1$

The remainder of the proof is dedicated to proving that any such network contains $3s - 3$ gates directly, i.e without recourse to the inductive hypothesis.

We introduce some further terminology and notation.

D_i denotes the unique successor of C_i , for $i \in S$. Let

$$C = \{ C_i : i \in S \} \quad ; \quad D = \{ D_i : i \in S \}$$

A *split* is a node v such that $\phi(v) \geq 2$. A path $(v \rightarrow w)$ is *free* in T if no node in the path (with the possible exception of v, w) is in C . A node, w in T , is a *free split* if:

FS1) w is a split.

FS2) There are distinct nodes u and v in T for which there are free paths $(u \rightarrow t)$ and $(v \rightarrow t)$ and w enters u and v .

A node w is a *collector* of free paths $(C_i \rightarrow t)$, $(C_j \rightarrow t)$ ($i \neq j$) if it is the first node common to both paths.

The next 4 properties of T were first given in Paul (1977).

Property 2: $\forall i \in S$ there exists a free path $(C_i \rightarrow t)$.

Proof: Suppose for some i in S there is no free path $(C_i \rightarrow t)$. Every path from C_i must go through some C_j ($j \neq i$). Since each C_j is \wedge -type one may construct an assignment, β , such that all variables, except x_i are fixed under β and:

$$\forall j \in S - \{i\} \quad \text{res}(C_j)^{\beta} \in \{0, 1\}$$

$(\alpha_1) = (\alpha_2) = i$, $(\alpha_3) = j \neq i$ and q is chosen so that:

$$f^{\beta} = (x_i)^e$$

for some $e \in \{0, 1\}$. T under β is independent of x_i but f^{β} is not. This contradiction proves the existence of a free path $(C_i \rightarrow t)$. \square

From Property(2) we have immediately that the sets C, D are disjoint.

Property 3: Let i, j be distinct elements of S and G be the collector of a free path $(C_i \rightarrow t)$ and a free path $(C_j \rightarrow t)$. Suppose there is no

free split on the paths $(C_i \rightarrow G)$, $(C_j \rightarrow G)$, except possibly G itself. Then

- i) G is an \oplus -type gate.
- ii) There is a free path $(C_i \rightarrow C_j)$ or a free path $(C_j \rightarrow C_i)$.

Proof: Suppose (i) is false and that G is an \wedge -type gate. Let β be the assignment which fixes each x_k , for $k \in S - \{i, j\}$, so that $res(C_k)^{\beta}$ is constant, $(\alpha_1) = k \neq i, j$, $(\alpha_2) = i$, $(\alpha_3) = j$ and q so that $x_{(\alpha_1)}^q = 1$. Then $f^{\beta} = x_i \oplus x_j$

(Note: The control variable is used here to guarantee that β exists. The proof of this property is one of the reasons for commencing the inductive step from values $s \geq 3$ since we are now assured of the existence of a suitable x_k . In this context see also Property(6) below).

With β a contradiction can be derived. For suppose $res(C_j)^{\beta}$ depends on x_i ; then we can set x_j to some c so that T under $\langle \beta, x_j := c \rangle$ is independent of x_i but $f^{\beta, x_j := c}$ is $x_i \oplus c$. An identical argument can be used if C_i depends on x_j . Now by the stated assumptions all paths $(C_i \rightarrow t)$ and $(C_j \rightarrow t)$ pass through some C_k , all of which are constant, or go through the collector G (since there are no free splits on the considered paths). But $res(G)^{\beta}$ computes an \wedge -type function of x_i and x_j or depends on only at most one variable. In either case t cannot realise the \oplus -type function required.

G is thus an \oplus -type gate. So suppose (ii) does not hold and there is neither a free path $(C_i \rightarrow C_j)$ nor a free path $(C_j \rightarrow C_i)$. Construct an assignment β for which $res(C_k)^{\beta}$ is constant for all k in $S - \{i, j\}$, $(\alpha_1) = i$, $(\alpha_2) = j$, $(\alpha_3) = k \in S - \{i, j\}$ and $q = 1$. Then, f^{β} realises an \wedge -type function of x_i and x_j and now employing a similar argument to that of (i) we derive the contradiction that $res(t)^{\beta}$ must

be an \wedge -type function, but such cannot be computed from the \oplus -type function of x_i, x_j given by G . \square

Property 4: $\forall i, j \in S (i \neq j) D_i \neq D_j$

Proof: Immediate from Property(3). \square

Properties (1) and (4) identify $2s$ distinct gates, $C \cup D$.

Property 5: T contains at least $s - 1$ distinct splits.

Proof: Let $S' = S$. Find i, j in S' for which there exist free paths $(C_i \rightarrow t)$ and $(C_j \rightarrow t)$ such that for all $k \in S' - \{i, j\}$ no free path $(C_k \rightarrow t)$ goes through the collector, G , of $(C_i \rightarrow t), (C_j \rightarrow t)$. Such a pair can always be found as follows:

Consider any pair of free paths $(C_i \rightarrow t)$ and $(C_j \rightarrow t)$ with collector G . Suppose some other free path $(C_k \rightarrow t)$ also goes through G . This path must intersect either the path $(C_i \rightarrow t)$ or $(C_j \rightarrow t)$ before G , by the definition of collector. Let H be the first gate at which this occurs and without loss of generality suppose H lies on the free path $(C_i \rightarrow t)$. Clearly H is the collector of the free path $(C_k \rightarrow t)$ and the free path $(C_i \rightarrow t)$ and is an ancestor of G . If i, k do not meet the condition required then the argument above can be repeated to select a new pair. The process must terminate since a gate preceding the current collector is always chosen at each stage and none of the gates in $D \cup C$ can be collectors.

From Property(4) one of the paths $(C_i \rightarrow G), (C_j \rightarrow G)$ splits into a free path to t or to C_j (resp. C_i), the split occurring at a gate preceding G . Without loss of generality suppose the path $(C_i \rightarrow G)$ is the one which splits. Now set $S' = S' - \{i\}$. Repeating the argument $s - 1$ times proves the result. The construction guarantees that no free path $(C_k \rightarrow t)$ intersects with the free path $(C_i \rightarrow G)$, and so no split is counted twice. \square

The remainder of proof is due to Blum.

The main idea is show that at least $(s-2)$ of the distinct $(s-1)$ splits identified above must be free splits. With Property(2) this will leave $2(s-2)+2$ wires which must be connected onto t , each wire lying on a free path to t . It will be shown that this entails the use of $s-3$ gates in addition to those in $C \cup D$, proving the lower bound $3s-3$.

First consider the $(s-1)$ splits located by Property(5) and suppose at most $(s-2)$ of these nodes are free splits. Let j be such that the split identified on the path from C_j is not free and i be such that no split on the path from C_i is found, i.e i, j are pessimal cases. By Property(3) i, j satisfy, with G the collector of the free paths $(C_i \rightarrow t)$ and $(C_j \rightarrow t)$.

- IJ1) There is no free split on the paths $(C_i \rightarrow G)$, $(C_j \rightarrow G)$, except possibly G .
- IJ2) G is \oplus -type.
- IJ3) There is a free path from C_i to C_j .

For this situation we have:

Property 6: $\forall k \in S - \{j\}$ there is a free path $(C_k \rightarrow C_i)$ or a free path $(C_k \rightarrow C_j)$.

Proof: A free path $(C_i \rightarrow C_j)$ has been identified by the choice of i . Suppose for some $k \in S - \{i, j\}$ there is neither a free path $(C_k \rightarrow C_i)$ nor a free path $(C_k \rightarrow C_j)$. Let β be the assignment which fixes all variables, excepting x_i, x_j, x_k so that $res(C_l)^{|\beta}$ is constant, for all l in $S' - \{i, j, k\}$, $(\alpha_1) = j$, $(\alpha_2) = i$, $(\alpha_3) = k$ and $q = 1$. Then:

$$f^\beta(x_i, x_j, x_k) = x_j \wedge (x_i \oplus x_k)$$

Now if $res(C_j)^{|\beta}$ is independent of x_i we can fix x_k to e in $\{0, 1\}$ so

that $\text{res}(C_k)^{| \beta, x_k = e}$ is constant and $f^{| \beta, x_k = e} = x_j \wedge x_i^{(e)}$. Since G is \oplus -type the argument used to prove Property(3) yields a contradiction. On the other hand, if $\text{res}(C_j)^{| \beta}$ does depend on x_i , then x_i may be fixed to some e so that $\text{res}(C_j)$ is constant. Note that since there is assumed to be no free path $(C_k \rightarrow C_j)$ or $(C_k \rightarrow C_i)$, $\text{res}(C_j)$ cannot depend on x_k . Again a contradiction results since after this T is independent of x_j but

$$f^{| \beta, x_i = e} = x_j \wedge x_k \quad \square$$

The final property will allow us to conclude that at most one of the splits identified is not a free split. In the statement below, j is again the path on which the non-free split is assumed to lie.

Property 7: $\forall k, l \in S - \{j\}$. If H is the collector of a free path $(C_k \rightarrow t)$ and a free path $(C_l \rightarrow t)$ then there is a free split, other than H , on the path $(C_k \rightarrow H)$ or the path $(C_l \rightarrow H)$.

Proof: (See Figure 7) Suppose there is neither a free split on the path $(C_k \rightarrow H)$ nor the path $(C_l \rightarrow H)$. From Property(6), using k and l instead of i and j , there is a path $(C_j \rightarrow C_l)$ or a path $(C_j \rightarrow C_k)$. But from the choice of j and i there are also paths $(C_l \rightarrow C_j)$ and $(C_k \rightarrow C_j)$ and this implies T contains a cycle. \square .

It is clear from Property(7) that T contains at least $s - 2$ distinct free splits. From this and Property(2) we have at least $2(s - 2) + 2$ wires on free paths which must be connected up to t . The gates in C cannot be used (by the definition of free path) and the gates in D can account for at most s wires. Thus at least $s - 2$ wires have to be connected using gates not in $C \cup D$. A single new gate can remove two wires but, if it is not the output, adds one. Hence $s - 3$ gates must be in T beside those in C and D . This gives:

$$\mathbf{C}(f) \geq |C| + |D| + s - 3 = 3s - 3 \quad \square$$

$\exists path (C_j \rightarrow C_k) \text{ or } (C_j \rightarrow C_l)$ (Property 6, with k, l instead of i, j)

$\exists path (C_k \rightarrow C_j) \text{ or } (C_l \rightarrow C_j)$ (Property 6, by the choice of i, j)

Figure 2.7

2.5) Some Upper Bounds on Combinational Complexity

We conclude this chapter by describing two network constructions which provide depth and size efficient realisations of some interesting classes of Boolean function. The first is a solution of the "parallel prefix problem", due to Ladner and Fischer (1980), which permits small depth and size simulations of finite state transducers by combinational networks. A consequence of this is a linear size, $O(\log n)$ depth network for computing the $n + 1$ bit sum of two n bit binary integers. The second construction presented is a linear size, $\log n$ -depth network capable of realising any n -input symmetric Boolean function, and is from Muller and Preparata (1975).

The existing literature concerned with combinational networks realising the basic arithmetic functions is extensive. However with the advent of more sophisticated technologies, entailing the assessment of new complexity metrics, it is perhaps debatable whether the solutions proposed are now of immediate, direct, practical significance. Given this and the technical complexity of the more important constructions, we will merely summarise the best of such results obtained to date at the the end of this section. The bibliographic notes following indicate further references.

Definition 2.5: Let $*$ be any associative, binary operation over some domain D . Let $\langle d_1, d_2, \dots, d_n \rangle$ be an n -tuple of variables taking values from D . The n -input *prefix problem* for $*$, is to compute the n products:

$$\{d_1 * d_2 * d_3 * \dots * d_i \mid 1 \leq i \leq n\}$$

A $*$ -*product network*, or simply product network, is a directed acyclic graph containing n input nodes associated with variables, $\langle d_1, \dots, d_n \rangle$ and $*$ -nodes, these having in-degree 2. We use \mathbf{C} , \mathbf{D} to denote the number of $*$ -nodes in (resp. depth of) a product network, S . •

In the obvious way any product network computes some set of products at its output nodes. When $D = \{0, 1\}$, the product network is just a Boolean network over the basis $\{*\}$, where $*$ may be one of $\{\wedge, \vee, \oplus, \iff\}$.

We wish to construct product networks to solve the prefix problem in minimal depth and size. The following construction of Ladner and Fischer (1980) presents a family $[P_k(n)]$ of product networks such that: $\forall 0 \leq k \leq \lceil \log n \rceil, \forall n \geq 1$:

$P_k(n)$ solves the n -input prefix problem.

$$\mathbf{C} \leq 2(1 + 2^{-k})n - 4$$

$$\mathbf{D} \leq k + \lceil \log n \rceil$$

Below we use m to denote $\lceil n/2 \rceil$. If $n = 1$ then $P_k(n)$ (i.e. $P_0(1)$) is just a single input node. For $n > 1$, $P_k(n)$ is formed recursively as follows:

$k = 0$: $P_0(n)$ consists of a copy of $P_1(m)$ and a copy of $P_0(n - m)$.

$$\langle d_1, d_2, \dots, d_m \rangle \text{ and } \langle d_{m+1}, \dots, d_n \rangle$$

denote the inputs of $P_1(m)$ and $P_0(n - m)$ respectively. Similarly let:

$$\langle y_1, \dots, y_m \rangle \text{ and } \langle y_{m+1}, \dots, y_n \rangle$$

denote the output nodes of these networks. $P_0(n)$ is formed by adding $n - m$ new * nodes, $\langle g_{m+1}, \dots, g_n \rangle$; the inputs of g_i being the output y_m of $P_1(m)$ and the output y_i of $P_0(n - m)$. The outputs $\langle p_1, \dots, p_n \rangle$ of $P_0(n)$ are then

$$\langle y_1, \dots, y_m, g_{m+1}, \dots, g_n \rangle$$

$k > 0$: $P_k(n)$ consists of a single copy of $P_{k-1}(m)$. If n is odd the inputs to this are the products:

$$\langle d_1 * d_2, d_3 * d_4, \dots, d_{n-2} * d_{n-1}, d_n \rangle$$

If n is even the inputs are:

$$\langle d_1 * d_2, d_3 * d_4, \dots, d_{n-1} * d_n \rangle$$

These being computed using $\lfloor n/2 \rfloor$ new *-nodes. Let $\langle y_1, y_2, \dots, y_m \rangle$ denote the outputs of $P_{k-1}(m)$. If n is odd the outputs of $P_k(n)$ are given by:

$$\langle d_1, y_1, y_1 * d_3, y_2, y_2 * d_5, \dots, y_{m-2}, y_{m-2} * d_{n-2}, y_{m-1}, y_m \rangle$$

whereas if n is even the outputs of $P_k(n)$ are computed by:

$$\langle d_1, y_1, y_1 * d_3, y_2, y_2 * d_5, \dots, y_{m-1}, y_{m-1} * d_{n-1}, y_m \rangle$$

The construction is depicted below:

Figure 2.8 (a)

Lemma 2.8: $\forall n \geq 1, \forall 0 \leq k \leq \lceil \log n \rceil$

- i) $P_k(n)$ solves the n -input prefix problem.
- ii) $\mathbf{D} \leq k + \lceil \log n \rceil$
- iii) $\mathbf{C} \leq 2(1 + 2^{-k})n - 4$

Proof:

- i) By induction on the value $2n + k$ ($k \geq 0, n \geq 1$). The base, $n = 1$ and $k = 0$, is obvious. Assume $P_k(n)$ solves the prefix problem

Figure 2.8 (b)

whenever $2n + k \leq s - 1$. To prove $P_k(n)$ solves it when $2n + k = s$ we distinguish two separate cases.

- A) $k = 0$: Since $n > 1$, we have $2m + 1 < 2n$ so from the inductive hypothesis both $P_1(m)$ and $P_0(n - m)$ are correct. The case $k = 0$ is now easily verified from the definition of $P_0(n)$, cf Fig(2.8)(a).
- B) $k > 0$: Again $P_{k-1}(m)$ is correct from the inductive hypothesis and using the definition of $P_k(n)$ cf Fig(2.8)(b)

this network solves the n -input prefix problem.

- ii) The depth bound is a straightforward induction using the easily obtained result that the n 'th output of $P_k(n)$ is always at depth $\lceil \log n \rceil$. The details are left to the reader.
- iii) (Outline) The description of $P_k(n)$ yields a system of recurrence relations. Solving these yields the claimed upper bound on \mathbf{C} . \square

One of the significant points of interest regarding this solution to the prefix problem is how it can be applied to give an efficient simulation of finite state transducers by combinational networks. We assume some familiarity with the basic concepts of automata theory. A *finite state transducer* (also known as a Mealy machine) is a quintuple $M = \langle Q, \Sigma, \Delta, \delta, \gamma \rangle$; Q a finite set of states; Σ a finite input alphabet; Δ a finite output alphabet (taken to be $\{0, 1\}$); $\delta : Q \times \Sigma \rightarrow Q$ the state transition function; $\gamma : Q \times \Sigma \rightarrow \Delta$ the output function. A state q_0 in Q is distinguished as the initial state. Given some input sequence $a_1 \cdots a_n$ in Σ^n , the output string $b_1 \cdots b_n$ generated by M is given by:

$$b_1 = \gamma(q_0, a_1)$$

$$b_2 = \gamma(\delta(q_0, a_1), a_2)$$

$$b_3 = \gamma(\delta(\delta(q_0, a_1), a_2), a_3) \quad \text{etc}$$

The computation proceeds sequentially. Let us now consider a parallel entity, combinational networks, which mimic the behaviour of such machines.

For each $e \in \Sigma$ define a function $M_e : Q \rightarrow Q$ by:

$$qM_e = \delta(q, e)$$

(That the argument of M_e is to the left is a notational convenience introduced for reasons which will become apparent below).

For any given input sequence $a_1 a_2 \cdots a_n$, the state of M after reading i symbols is clearly:

$$q_0 M_{a_1} \circ M_{a_2} \circ \cdots \circ M_{a_i}$$

(\circ denoting functional composition).

A combinational network which produces the output sequence $b_1 \cdots b_n$ and the final state from a given input $a_1 \cdots a_n$ and the initial state q_0 can be built in four stages.

S1) Compute $M_{a_1}, M_{a_2}, \dots, M_{a_n}$.

S2) Compute for each $1 \leq i \leq n$ the functions N_i , where:

$$N_i = M_{a_1} \circ M_{a_2} \circ \cdots \circ M_{a_i}$$

S3) Compute for each $1 \leq i \leq n$, the next state $q_i = q_0 N_i$.

S4) Finally compute the outputs b_i , using $b_i = \gamma(q_{i-1}, a_i)$.

We now give a detailed description of how these steps may be realised.

Each input a_i is represented by a binary word containing $r = \lceil \log |\Sigma| \rceil$ bits. Each state q_i is represented by a binary $|Q|$ -tuple $\langle s_0, \dots, s_{|Q|-1} \rangle$ in which $s_i = 1$ and all other elements are 0.

Each function M_e , for $e \in \Sigma$ is encoded by a $|Q| \times |Q|$ Boolean matrix, T_e ; the (i, j) entry of this matrix being 1 if and only if $\delta(q_i, e) = q_j$. With this convention, evaluation of qM_e can be performed as a vector matrix product, viz $\underline{s} T_e$, where \underline{s} is the encoding of state q . Also the "functional composition" $M_e \circ M_f$ reduces to the multiplication of 2 $|Q| \times |Q|$ Boolean matrices (using \vee instead of $+$, \wedge instead of \times). Thus:

$$qM_e \circ M_f = \underline{s} T_e T_f$$

Since matrix multiplication is associative, so stage (S2) can be implemented using the solution to the prefix problem.

We can now construct a combinational network to compute (S1-S4); the total number of input *variables* is nr , \mathbf{x}_i denotes the r -tuple of variables corresponding to the transducer input a_i .

T1) For each \mathbf{x}_i the appropriate matrix T_{a_i} must be selected, thus some function:

$$T: \{0, 1\}^r \rightarrow \{0, 1\}^{|\mathcal{Q}|^2}$$

is being computed. The (p, q) entry of T_{a_i} is given by:

$$\bigvee_{e \in \Sigma} \delta(d_e(\mathbf{x}_i) \wedge T_e[p, q])$$

where d_e is the "equivalence function" of Sect(2.2) and $T_e[p, q]$ denotes the (p, q) entry of T_e . Thus the matrix T_e is selected if and only if the input corresponds to e in Σ .

No more than $r \cdot 2^r - 1$ gates are needed to compute a single matrix entry. Thus:

$$n |\mathcal{Q}|^2 (r \cdot 2^r - 1)$$

gates suffice to compute all the outputs of (T1).

T2) The second stage is realised by the prefix network $P_k(n)$, constructed earlier, in which the product nodes realise $|\mathcal{Q}| \times |\mathcal{Q}|$ Boolean matrix multiplication. From Lemma(2.5) at most $4n - 4$ of these are needed, if $k = 0$, (fewer for larger fixed k). Assuming the obvious matrix product network, this stage entails the use of an additional:

$$(4n - 4) (2|\mathcal{Q}|^3 - |\mathcal{Q}|^2) \text{ gates.}$$

- T3) This is just a vector-matrix product, however since q_0 is encoded as $\langle 1, 0, 0, \dots, 0, 0 \rangle$ this stage involves no extra gates. Only the appropriate entries from each matrix computed in (T2) need be passed to
- T4) in which q_1, \dots, q_n denote the encoded states selected. Each output b_i can now be computed from the expression:

$$\bigvee_{q \in Q} \bigvee_{e \in \Sigma} (d_q(q_i) \wedge d_e(\mathbf{x}_i) \wedge \gamma(q, e))$$

Thus a total:

$$n \lceil 2^r |Q| (|Q| + r) - 1 \rceil \text{ gates}$$

suffice to compute all the b_i .

Summing the contributions of (T1), (T2) and (T4) gives at most $c_1 n - c_2$ gates, for some constants c_1, c_2 . Similarly the depth of the construction can be shown to be no more than $c_3 \lceil \log n \rceil$ for some constant c_3 depending on M .

The multiplicative constants in the size and depth bounds may often be substantially smaller for specific simulations, since one may be able to utilise a clever encoding of the state and composition functions. A good example of this, given by Ladner and Fischer, is the derivation of a combinational network realising binary addition, the function $ADD(\mathbf{X}_n, \mathbf{Y}_n)$ of Sect(1.2).

Consider the 2 state transducer depicted in Fig(2.10)

The output gives the result of summing the 2 input bits and any carry from the previous addition. This carry being stored as the label of the current state. Since the behaviour on input 10 is identical to that on input 01 there are only 3 functions, over the states 0 and 1, to consider: M_{00} , M_{01} and M_{11} . From inspection of Fig(2.10) it may be

Figure 2.9

verified that:

$$\forall i, j, k, l \in \{0, 1\} \quad M_{ij} \circ M_{kl} \in \{M_{00}, M_{01}, M_{11}\}$$

i.e the set of functions is closed under composition.

Following Ladner and Fischer (1980), instead of representing M_{ij} for $ij \in \{00, 01, 10, 11\}$ by a 2×2 Boolean matrix, it may be encoded by 2 bits: r (result) and c (carry), thus:

Figure 2.10

Input:	00	01	10	11
rc	00	01	01	10

Table 2.1

So for input xy , $r = x \wedge y$ and $c = x \oplus y$. The result, rc , of composing two function r_1c_1 and r_2c_2 is then given by Table(2.2). So that: $r = r_2 \vee (r_1 \wedge c_2)$ and $c = c_1 \wedge c_2$.

Finally the state which results by applying M_{ij} to a given state, $s \in \{0, 1\}$ is shown in Table(2.3); the output resulting in state t on input corresponding to function rc is given in Table(2.4).

o	r_2c_2			r_1c_1
	00	10	10	
00	00	00	10	
01	00	01	10	
10	00	10	10	

Table 2.2

		$Function = rc$			
		sM	00	01	10
State	0	0	0	1	
	s	0	1	1	

Table 2.3

		$Input = rc$		
		00	01	10
State	0	0	1	0
	t	1	0	1

Table 2.4

In Table(2.3) the result is given by the expression $r \vee (s \wedge c)$; In Table(2.4) the output is given by $t \oplus c$.

We can now prove:

Theorem 2.20:

$$\mathbf{C}(\text{ADD}(\mathbf{X}_n, \mathbf{Y}_n)) \leq 3\mathbf{C}(P_k(n)) + 2n \quad (2.20)$$

$$\mathbf{D}(\text{ADD}(\mathbf{X}_n, \mathbf{Y}_n)) \leq 2D(P_k(n)) + 2 \quad (2.21)$$

and both bounds are achievable simultaneously.

Proof: The basic template (S1-S4) above, is implemented by the stages (A1-A4) below.

- A1) The function, $r_i c_i$ associated with each input pair $x_i y_i$ is encoded by computing $\langle x_i \wedge y_i, x_i \oplus y_i \rangle$, following Table(2.1). This uses $2n$ gates and requires depth 1.
- A2) The computation of N_1, \dots, N_n from the input pairs $r_1 c_1, \dots, r_n c_n$ proceeds using the prefix network $P_k(n)$, in which product nodes realise the functions r, c according to Table(2.2). This can be done with $3C(P_k(n))$ gates and in depth $2D(P_k(n))$.
- A3) By choosing the initial state to be state 0, the next state computation merely selects the r component of each output pair in (A2) (cf. Table(2.3)). Thus this stage adds no extra gates or depth.
- A4) The n outputs are computed using Table(2.4); the final state, i.e last carry gives the $n + 1$ output. This requires in total at most n gates and extra depth 1.

Summing each contribution proves the theorem. \square

The multiplicative constants are bettered in the construction of Khrapchenko (1967); these bounds are given at the end of this section.

The method for efficiently realising symmetric functions given in Muller and Preparata (1975) also involves a form of addition network; one which calculates the total number of 1's present in an assignment to \mathbf{X}_n .

Theorem 2.21: (Muller and Preparata, 1975) $\forall f \in S_n$

$$\mathbf{C}(f) \leq 5n + O\left(\frac{n}{\log n}\right)$$

$$\mathbf{D}(f) \leq 7\lceil \log n \rceil$$

Proof: We shall assume that $n = 2^m - 1$ for some $m \geq 1$. The construction is in two stages: the first computes the m -output function WT which is defined by:

$$WT = \mathbf{d} = \langle d_{m-1}, d_{m-2}, \dots, d_0 \rangle$$

where \mathbf{d} is the binary representation of the total number of 1's assigned to \mathbf{X}_n , d_{m-1} being the most significant digit, d_0 the least.

Since any symmetric function is completely specified by its spectrum, it may be regarded as a function $g(\mathbf{d}): \{0, 1\}^m \rightarrow \{0, 1\}$, the assignments to \mathbf{d} corresponding to the number of 1's in assignments to \mathbf{X}_n . In this way, given a realisation of WT , any symmetric function can be computed from the m -tuple of outputs \mathbf{d} using some combinational network. The second stage consists of an appropriate network to perform this task. From Thm(2.7) and Thm(2.10) such a network can be constructed to have size $\frac{2^m}{m} = \frac{n}{\log n}$ or depth $m + 1$. It is therefore sufficient to prove that the first stage may be built using at most $5n$ gates and with depth $6m$.

The construction is recursive. Let $k = 2^{m-1} - 1 = (n - 1)/2$. Suppose S_1 is a network computing $WT(x_1, \dots, x_k)$ at outputs $\langle a_{m-2}, \dots, a_0 \rangle$, and that S_2 realises $WT(x_{k+1}, \dots, x_{n-1})$ at outputs $\langle b_{m-2}, \dots, b_0 \rangle$. The required output for WT is then just the result of adding these $2(m - 1)$ -tuples together with x_n .

For this addition a chain of $m - 1$ *full adders* (FAs) is used. An FA has 3 inputs y, z, c and 2 outputs; r giving the result of adding the single bits y, z and c (the carry forward); and nc which is the next carry forward. So:

$$r = y \oplus z \oplus c \quad ; \quad nc = c \wedge (y \oplus z) \vee y \wedge z$$

An FA can be realised using 5 gates and depth 3. The chain is illustrated in Figure(2.11). The inputs to the i 'th block are a_i, b_i and c_{i-1} (the previous carry, with c_{-1} taken to be x_n). The outputs are d_i as result, and c_i .

So the combinational complexity of $WT, S(n)$, satisfies the relation:

$$S(n) \leq 5(m - 1) + 2S((n - 1)/2) \leq 5(n - m - 1)$$

By an easy induction on m .

The depth bound is only slightly more difficult. Let \mathbf{D} denote the depth at which the i 'th output d_i is computed when realising WT , for $n = 2^m - 1$. We shall prove:

$$\mathbf{D} \leq 4m + 2i + 1 \quad \forall m, \forall 0 \leq i \leq m$$

The base cases $m = 0$ and $m = 1$ are trivial. Assume that $\mathbf{D} \leq 4m' + 2i + 1$ for all $m' < m$. Using the recursive construction above and this hypothesis it follows that each output a_i, b_i is at depth at most $4(m - 1) + 2i + 1$. We additionally assume inductively that the carry in to the i 'th FA, c_{i-1} , is computed at depth no more than $4(m - 1) + 2i + 3$. With $c_1 = x_n$ the base here is again trivial. We now have that d_i requires additional depth only 2, hence:

$$\mathbf{D} \leq 4(m - 1) + 2i + 5 = 4m + 2i + 1$$

The inductive step for the depth c_i follows in a similar manner. Now

Figure 2.11

choosing $i = m$ yields the claimed bound on depth. \square

The theorem below summarises the best upper bounds on size and depth for the arithmetic operations addition, multiplication and integer division. In all cases the bounds are simultaneously achievable.

Theorem 2.22:

$$\mathbf{C}(\text{ADD}(\mathbf{X}_n, \mathbf{Y}_n)) \leq 9n \quad (2.22)$$

$$\mathbf{D}(\text{ADD}(\mathbf{X}_n, \mathbf{Y}_n)) \leq \log n + o(\log n)$$

$$\mathbf{C}(\text{MULT}(\mathbf{X}_n, \mathbf{Y}_n)) = O(n \log n \log \log n) \quad (2.23)$$

$$\mathbf{D}(\text{MULT}(\mathbf{X}_n, \mathbf{Y}_n)) = O(\log n)$$

$$\mathbf{C}(\text{DIVN}(\mathbf{X}_n, \mathbf{Y}_n)) = O(n^k) \quad \text{for some fixed } k \quad (2.24)$$

$$\mathbf{D}(\text{DIVN}(\mathbf{X}_n, \mathbf{Y}_n)) = O(\log n)$$

Proof: (2.22) is from Khrapchenko (1967); (2.23) from Schonhage and Strassen (1971); (2.24) is proved in Beame, Cook and Hoover (1984). \square

Bibliographic Notes

The complexity of computing Boolean functions on various automaton models has been considered by Breitbart (1968) and Sholomov (1970). Alt (1984) and Jung (1985) examine depth efficient simulations of arithmetic networks by Boolean networks. The properties of $\mathbf{C}(B_n)$, usually referred to as the *Shannon function* by Soviet authors, have been studied in Karpova (1975) and Orlov (1971). Several researchers have investigated different forms of "universal" network. Preparata and Muller (1970) show that every function in B_n can be obtained as a subfunction of one in B_{n+m} and obtain an asymptotically minimal bound on the size of m . Valiant (1976) constructs a network of size $O(c \log c)$ and depth $O(c)$ which can simulate any network of size c given a suitable setting of its control inputs. In similar vein

Cook and Hoover (1985) present a network of size $O\left(\frac{c^3 d}{\log c}\right)$ and depth $O(d)$ which can simulate any network of size c and depth d . Lupanov (1958) also gives upper bounds for the complexity of all sets of functions in $B_{n,m}$ for m satisfying certain conditions. These are mentioned in the following chapter.

Besides those given in Sect(2.4) a number of other linear lower bounds on combinational complexity are known. Red'kin (1973) appears to be the first detailed presentation of an inductive gate elimination argument, although this is only for the basis $\{\wedge, \vee, \neg\}$. Schnorr (1976b) gives exactly matching upper and lower bounds on the combinational complexity of the function:

$$\bigwedge_{i=1}^{n-1} (x_i \iff x_{i+1})$$

Linear lower bounds are also given by Bremer (1974), Harper (1975) and Harper, Hsieh and Savage (1975).

As we mentioned previously the literature covering the combinational complexity of arithmetic functions is substantial. Addition networks being studied in Avizienis (1961), Sklansky (1960a,1960b) and Spira (1973); Earlier results on multiplication were found by Karatsuba and Ofman (1962), Ofman (1962), Toom (1963) and Wallace (1964).

Paul (1975) and Ulig (1974) consider the complexity of realising functions on disjoint sets of variables; both papers showing that combinational complexity is not additive (i.e there exist functions $f(\mathbf{X}_n)$, and $g(\mathbf{Y}_n)$ for which $\mathbf{C}(f \vee g) < \mathbf{C}(f) + \mathbf{C}(g) + 1$). Pippenger (1977) and Sholomov (1971) examine the application of information theory as a means of obtaining complexity bounds. Blum and Seysen (1984) consider simultaneous computation of \wedge and $\neg\vee$.

Finally, Yablonskii (1959a,b) and Nigmatullin (1984,1985) attempt to account for the fact that proving superlinear lower bounds on network complexity is difficult.

Chapter 3

Monotone Network Complexity

... it was clear that the end was still far, far off, and that the hardest and most complicated part was only just beginning.

Anton Chekhov **The Lady with the Dog**

In this and the remaining chapters our primary concern is with a number of restricted models of Boolean network: monotone; formulae; bounded-depth; and planar. There are several important general differences between monotone networks and these other models. The final three all employ graph-theoretic restrictions: formulae compel gates to have out-degree at most 1; bounded-depth networks permit arbitrary fan-in gates of certain types, but limit the depth to being constant; the planar model requires the underlying graph to be planar. Additionally all of these turn out to be functionally complete in the sense that each may realise any f in B_n . Neither is true of the monotone model; this involving a restriction of content, the basis, rather than of form, the graph structure.

That one considers simplified models is largely due to the lack of progress in developing powerful lower bound arguments pertinent to combinational networks; the hope being that these restricted forms will prove more amenable to analysis. With respect to arbitrary networks the aims of such models are twofold: to gain insight into proof techniques for combinational networks via lower bound methods for the restricted model; and to determine if such networks may efficiently simulate unrestricted networks.^{a)} Thus formulae are of interest

a) The assertion here requires some qualification in the case of bounded-depth networks,

since it is possible to deduce bounds on network depth from bounds on formula size, cf Theorem(2.4); the bounded-depth model includes a class of networks whose complexity exactly corresponds with the number of products in the minimal DNF for the computed function. Finally the planar network complexity of f can be shown to be at most $C(f)^2$, so large enough bounds in this model yield superlinear bounds on combinational complexity.

For the class of monotone Boolean networks there *appears* to be no such strong theoretical motivation. However a number of significant results, derived from 1983 onwards, have shown that it may now be reasonably contended that this model offers the greatest potential for obtaining realistic complexity bounds. For this reason and the fact that the other models mentioned above may themselves be restricted to monotone instances, monotone Boolean networks will be the first restricted model which will be examined in detail.^{b)}

Prior to 1983 the study of monotone network complexity was motivated in a number of ways. The set of n -input monotone Boolean functions, M_n , has long been of historical interest in certain areas of algebra and combinatorial mathematics, dating back to the work of Dedekind in the late 19th century. Despite the fact that the monotone basis $\{\wedge, \vee\}$ is incomplete, a great many computationally interesting functions are monotone, e.g the threshold functions and a large number of NP -complete problems. Even in cases where a function is not monotone it is often possible to consider instantiations which are, e.g the set of Boolean functions defining integer multiplication is not in

which have mainly been investigated because of their relevance to a question concerning the separation of particular complexity classes. This is discussed in greater detail in Chapter(5).

b) We could of course define other compositions of restriction, e.g bounded-depth formulae. In practice such models have little theoretical and even less pragmatic significance.

$M_{2n,2n-1}$, but a special case is Boolean convolution, a set of monotone functions. The progress made in obtaining good techniques for other monotone models provides another justification. The complexity of monotone arithmetic networks i.e with only the operations $+$, \times permitted was considered in both Schnorr (1976c) and (Jerrum and Snir, 1982). The former derives exponential lower bounds on the number of additions required to compute certain rational functions, bounds which are sometimes exact; the latter similar results for the number of multiplications necessary. Lingas (1979) proves similar bounds in other monotone models of computation.

This chapter presents a detailed and extensive account of the theory of monotone network complexity. In Section(3.1) we describe the history of a classical problem first formulated in Dedekind (1897), namely to determine $|M_n|$ (denoted $\psi(n)$ henceforward). A lower bound allows complexity bounds for almost all monotone functions to be obtained using Shannon's argument. The upper bound on $\psi(n)$ proved in Hansel (1966) is also given. Although Hansel's result is not quite optimal, the bounds of Kleitman (1969), Kleitman and Markowsky (1974) and Korshunov (1981) all improving it, the concepts introduced in its derivation will be useful subsequently. This section continues with two upper bounds on the network complexity of all monotone Boolean functions. The first, also from Hansel (1966), is an asymptotically optimal construction for *combinational* networks realising functions in M_n . This utilises a powerful design method detailed in Lupanov (1961b, 1965b), known as the "Principle of Local Coding". The second construction, described in Red'kin (1979), concerns the computation of monotone functions using only the monotone basis $\{\wedge, \vee, 0, 1\}$. We conclude this section with a minor improvement to the complexity hierarchy results of Thm(2.11) for monotone networks.

Section(3.2) is a prelude to a number of the lower bound results presented later in the chapter. It introduces an important tool for reasoning about the optimality of monotone networks, which was originally developed in Paterson (1975) and (Mehlhorn and Galil, 1976). This is the concept of replacement rules. The application of this technique is discussed and some characterisation theorems from Dunne (1984c) are proved.

The earliest indications that monotone networks are a tractable model in which to obtain good complexity results came in the mid 1970's with the appearance of the first superlinear lower bounds on the complexity of *sets* of monotone Boolean functions, i.e members of the class $M_{n,m}$. Section(3.3) deals with a number of these results, among them the precise description of networks computing Boolean matrix product from Paterson (1975); the result of Weiss (1983) on Boolean Convolution; and a bound on the complexity of sets of Boolean sums from Mehlhorn (1979). Refinements of the inductive gate elimination approach, all these results utilise replacement rule arguments in some way.

The following two sections consider the complexity of single output functions, Section(3.4) giving some linear lower bounds on the complexity of threshold functions from Dunne (1984b, 1985a) and a particularly elegant $4n$ lower bound of Tiekenheinrich (1984). An upper bound on the monotone network complexity of all fixed threshold functions is also given.

Thus far there is but little to merit the status we have earlier accorded this model. It is in the results of the concluding sections of this chapter that its significance is confirmed. In 1985 the Soviet mathematician Razborov obtained the first *superpolynomial* lower bounds on monotone network complexity. Razborov (1985a, 1985b) proves such results for a range of "natural" problems in graph theory.

The method employs a startlingly innovative combinatorial approach and may be phrased to yield a general inequality on monotone function complexity. These results represent a tremendous advance and yet were improved, by different methods, in the paper of Andreev (1985). Andreev derived exponential lower bounds, for an alternative class of monotone functions. Independently of Andreev, Alon and Boppana (1986) showed that certain combinatorial arguments of Razborov (1985a) could be sharpened^c. In doing this they improved Razborov's results to exponential and exceeded the best lower bound obtained by Andreev. In Section(3.5) a complete account of these techniques is presented.

In Section(3.6) the relation between monotone and combinational complexity is investigated by introducing the concepts of "standard circuit" and "pseudo-complement". An important result of Berkowitz (1983) is proved, this showing that superlinear lower bounds on the combinational complexity of *all* functions in $M_{n,m}$ follow directly from sufficiently large lower bounds on the *monotone* complexity of a related class of functions, called "slice functions". The properties of slice functions have been investigated further in Wegener (1985, 1986) and Dunne (1984a, 1985c, 1986). We conclude this chapter by describing some of the results of these papers.

In total the results of Section(3.5) show that large lower bounds on monotone complexity can be derived and, as will be apparent, by two quite general methods. The results of Section(3.6) demonstrate that if these, or other methods, can be adapted to apply to certain types of monotone function then a general technique for achieving good lower bounds on combinational complexity has become

c) We note here that Razborov (1985a) gives no proofs, only an outline of the method, thus Alon and Boppana had effectively to derive these results directly. Razborov (1985b) does give a detailed description of his approach including all proofs.

available. It is these facts that establish the importance of monotone network theory.

Subsequently we shall use $\mathbf{C}^m(f)$ to denote the monotone network complexity of a function $f \in M_n$, and $\mathbf{C}^m(S)$ to denote the size of a monotone network S .

3.1) Bounds for almost all monotone Boolean functions

A consequence of Corollary(2.2) is that a lower bound on the combinational (and so also monotone) complexity of "almost all" functions in M_n can be obtained from a lower bound on $\psi(n) = |M_n|$. The problem of exactly determining this quantity for arbitrary n was first raised by Dedekind (1897). It is not difficult to derive a crude lower bound on $\psi(n)$ by reasoning as follows:

Let E_n denote the binomial coefficient $\binom{n}{\lfloor n/2 \rfloor}$. From Lemma(1.1) any prime implicant of $f \in M_n$ is just the product of some subset of its formal arguments. Clearly there are exactly E_n different products of $\lfloor n/2 \rfloor$ variables. With this we can identify at least 2^{E_n} distinct functions in M_n , since any subset of these E_n products can be interpreted as the set of prime implicants of some function in M_n and no two different subsets define the same function.

As a result of this we have:

Theorem 3.1: For almost all $f \in M_n$, for all $\varepsilon > 0$ and n sufficiently large:

$$\mathbf{C}^m(f) \geq \mathbf{C}(f) > \sqrt{\frac{2}{\pi}} \frac{(1 - \varepsilon)2^n}{n^{3/2}}$$

Proof: Using Stirling's approximation $E_n \sim \sqrt{\frac{2}{\pi n}} 2^n$ and Corollary(2.2). \square

At present, no concise closed form for $\psi(n)$ has been found, and the exact value of this function is not known for n above 6. A considerable body of work exists concerning the asymptotic behaviour of $\psi(n)$. The list following summarises the history of Dedekind's problem.

- 1) Dedekind (1897) proves $\psi(3) = 20$, $\psi(4) = 168$.
- 2) Church (1940) demonstrates that $\psi(5) = 7581$.
- 3) Ward (1946) obtains the result $\psi(6) = 7,828,354$.
- 4) Gilbert (1954) considers the behaviour of $\log \psi(n)$ and proves:

$$E_n \leq \log \psi(n) \leq E_n \log n$$

- 5) Korobkov (1963) strengthens the upper bound of Gilbert (1954) to

$$\log \psi(n) \leq \frac{3 \log 3}{(3^{2/3} - 1)^{3/2}} E_n$$

- 6) Hansel (1966) improves Korobkov (1963),

$$\psi(n) \leq 3^{E_n}$$

- 7) Kleitman (1969) further reduces the upper bound on $\psi(n)$,

$$\log \psi(n) \leq \left(1 + O\left(\frac{\log_e n}{\sqrt{n}}\right)\right) E_n$$

- 8) Kleitman and Markowsky (1974) strengthen the lower bound of Gilbert (1954) and this upper bound,

$$(1 + O\left(\frac{1}{2^{\sqrt{n}}}\right))E_n \leq \log \psi(n) \leq (1 + O\left(\frac{\log_e n}{n}\right))E_n$$

- 9) Korshunov (1981) derives asymptotically matching upper and lower tolerances for $\psi(n)$. For n even (resp. odd),

$$\psi(n) \sim 2^{E_n} \exp\left(\binom{n}{\frac{n}{2}-1} (2^{\frac{-n}{2}} + n^2 2^{-n-5} - n 2^{-n-4})\right)$$

$$\psi(n) \sim 2^{1+E_n} \exp(G(n))$$

where $G(n)$ is

$$\left(\binom{n}{\frac{n-3}{2}} (2^{\frac{-n-3}{2}} - n^2 2^{-n-6} - n 2^{-n-3})\right) + E_n (2^{\frac{-n-1}{2}} + n^2 2^{-n-4})$$

Korshunov's result estimates $\psi(6)$ as 7,996,118; an error of just 2% using the result of Ward (1946).

The last 3 results mentioned above are too lengthy to present here. Instead we shall be content to derive Hansel's upper bound. Both Kleitman (1969) and (Kleitman and Markowsky, 1974) are in effect improvements to the basic method presented in Hansel (1966). Korshunov (1981) introduces radically different ideas, some of which will be mentioned below in the context of complexity hierarchies for monotone Boolean functions.

Hansel's proof is based on the properties of a widely studied partition of $2^{\mathbf{X}_n}$ into E_n connected symmetric *chains*. A connected, symmetric chain in $2^{\mathbf{X}_n}$ being a totally ordered collection of subsets,

$$R_{\frac{n}{2}-j} \subset R_{\frac{n}{2}-j+1} \subset \cdots \subset R_{\frac{n}{2}+j}$$

the set R_i containing exactly i elements of \mathbf{X}_n . In describing this we shall employ the characterisation presented in Greene and Kleitman (1976). It will be convenient to regard subsets of \mathbf{X}_n as defining monoms, for which we use the partial order \leq as before. Thus if P and Q are two subsets of \mathbf{X}_n with $P \subseteq Q$ then the corresponding monoms, p and q , satisfy $q \leq p$. We shall retain this convention of using upper case Roman letters to denote subsets and lower case for the implied monom. The notation $|p|$ is employed as a shorthand for $|\text{var}(p)| = |P|$.

Definition 3.1: Let \underline{w} be any finite binary string. \underline{w} is *well-formed* if and only if

W1) \underline{w} is the empty string.

or

W2) $\underline{w} = \underline{w}_1 \underline{w}_2$ and $\underline{w}_1, \underline{w}_2$ are well-formed,

or

W3) $\underline{w} = 1 \underline{w}_1 0$ and \underline{w}_1 is well-formed.

Let $\underline{w} = w_1 w_2 \cdots w_n$ be a binary string of length n . j is a *free 0* if there is no i ($1 \leq i < j$) for which the substring $w_i w_{i+1} \cdots w_j$ of \underline{w} is well-formed. Similarly j is a *free 1* if there is no i ($j < i \leq n$) for which the substring $w_j w_{j+1} \cdots w_i$ is well-formed.

(i, j) is a *bound pair* in \underline{w} if $i < j$ and the substring $w_i \cdots w_j$ of \underline{w} is well-formed. •

Informally well-formed strings correspond to balanced sequences of left "(" and right ")" parentheses, regarding 1's as left and 0's as right. Free 0's and 1's are then un-matched parentheses and bound pairs form matching brackets.

Fact 3.1: Let $\underline{w} = w_1 w_2 \cdots w_n$ be a binary word of length n .

- i) If i is a free 1 in \underline{w} then $\forall j > i$, j is not a free 0 in \underline{w} .
- ii) If i is a free 0 in \underline{w} then $\forall j < i$, j is not a free 1 in \underline{w} .

Proof: Since (ii) is immediate from (i), it is sufficient to prove (i) only. Suppose that i is a free 1 in \underline{w} but that (i) is false. Let $j > i$ be the free 0 in \underline{w} such that $j - i$ is minimal. The subword $w_{i+1} \cdots w_{j-1}$ cannot be well-formed for then (i, j) would be a bound pair in \underline{w} . So from the choice of j , this subword contains a free 1. Let k be the free 1, $i + 1 \leq k \leq j - 1$ such that $j - k$ is minimal. Now a contradiction results since $w_{k+1} \cdots w_{j-1}$ is well-formed and thus (k, j) is a bound pair in \underline{w} . \square

Any subset, P , of \mathbf{X}_n can be encoded as a binary string which is just the assignment to \mathbf{X}_n fixing exactly the variables in P to 1, e.g if $n = 5$ and $P = \{x_1, x_3, x_4\}$ then the encoding is 10110; here (1,2) and (4,5) are bound pairs and 3 is a free 1. $\beta(P)$ (or $\beta(p)$) will denote this encoding of an arbitrary subset (monom). The partition of $2^{\mathbf{X}_n}$ is formed by considering a relation *TIED*, defined between subsets of \mathbf{X}_n . For P and $Q \subseteq \mathbf{X}_n$, $\langle P, Q \rangle \in \textit{TIED}$ if $\beta(P)$ and $\beta(Q)$ contain exactly the same bound pairs, e.g for $n = 5$ $\langle 10010, 10110 \rangle \in \textit{TIED}$. Kleitman and Markowsky (1974) note that the properties of this relation have been studied and used by a large number of authors. The result below summarises several which will be of interest.

Fact 3.2: (Parts (v) and (vi) are due to Hansel)

- i) *TIED* is an equivalence relation.
- ii) Let $C = \{P_1, \dots, P_r\}$ be any equivalence class of *TIED*. Then: $P_i \subset P_{i+1}$, and $|P_i| = |P_{i+1}| - 1$, $\forall 1 \leq i < r$; i.e Each equivalence class is a chain, totally ordered by \subset .
- iii) *TIED* contains exactly E_n equivalence classes.

- iv) For any $P \subseteq \mathbf{X}_n$, the length of the chain C containing P (i.e $|C|$) is 1 plus the total number of free 0's and free 1's in $\beta(P)$.
- v) $\forall 0 \leq r \leq \lfloor n/2 \rfloor$, if $P_1 \subset P_2 \subset P_3$ is a consecutive sequence of 3 subsets of \mathbf{X}_n occurring in a chain of length $n - 2r + 1$ then the subset $P_1 \cup (P_3 - P_2)$ of \mathbf{X}_n occurs in a chain of length $n - 2r - 1$.
- vi) $\forall 0 \leq r \leq \lfloor n/2 \rfloor$, there are exactly:

$$\binom{n}{r} - \binom{n}{r-1} \text{ chains}$$

of length $n - 2r + 1$. (For $k < 0$ we take $\binom{n}{k}$ as 0.)

Proof:

- i) This is trivial and is left to the reader.
- ii) Suppose the contrary and that $\langle P, Q \rangle \in TIED$ but $P \not\subseteq Q$ and $Q \not\subseteq P$. The set of ordered pairs $P - Q \times Q - P$ must then be non-empty. Let $\langle x_i, x_j \rangle$ be a pair in this set such that $|j - i|$ is minimal. Without loss of generality we assume that $j > i$, $x_i \in P - Q$ and $x_j \in Q - P$. Thus $\beta(P)$, $\beta(Q)$ may be depicted as:

	1	2	...	$i-1$	i	$i+1$...	$j-1$	j	$j+1$
$\beta(P)$				1	<	\underline{z}	>	0	
$\beta(Q)$				0	<	\underline{z}	>	1	

From the choice of x_i and x_j the substring, \underline{z} , between positions $i + 1$ and $j - 1$ must be the same in $\beta(P)$ and $\beta(Q)$. i must be a free 1 in $\beta(P)$ and a free 0 in $\beta(Q)$, for otherwise $\langle P, Q \rangle \notin TIED$. From Fact(3.1)(i) it follows that j is not a free 0 in $\beta(P)$ and thus (k, j) is a bound pair in $\beta(P)$ for some $k < j$ (in fact $i < k < j$). But (k, j) cannot be a bound pair in $\beta(Q)$. This contradicts $\langle P, Q \rangle \in TIED$

and so $P \subseteq Q$ or $Q \subseteq P$, proving the first part of (ii).

For the second part consider any equivalence class, C , of *TIED*:

$$C = \{P_1, P_2, \dots, P_r\}$$

From our argument above C may be regarded as a chain, i.e. $P_i \subset P_{i+1}$, $\forall 1 \leq i < r$. It follows that $\beta(P_1)$ contains no free 1's for otherwise we could remove some element from P_1 and not affect the set of bound pairs.

Let

$$k = \max \{i \mid i \text{ is a free 0 in } \beta(P_1)\}$$

and $Q = P_1 \cup \{x_k\}$. Then $\langle P_1, Q \rangle \in \textit{TIED}$ since k could only form a bound pair with some free 0 at a position $> k$ in $\beta(Q)$ and none such exists. From the ordering of C , it must be the case that $Q = P_2$ and $|P_1| = |P_2| - 1$. An identical argument establishes that $|P_i| = |P_{i+1}| - 1$ for each $1 \leq i < r$ as claimed.

iii) From (ii) no 2 subsets of \mathbf{X}_n containing exactly $\lfloor n/2 \rfloor$ elements are in the same equivalence class, hence the number of classes is $\geq E_n$. On the other hand, suppose some class, C , contains no subset with exactly this many elements. Either the minimal subset, P of C , must contain at least $\lfloor n/2 \rfloor + 1$ members and $\beta(P)$ has no free 1's, or the maximal subset has at most $\lfloor n/2 \rfloor - 1$ members and no free 0's. But then in both cases there are at least $\lfloor n/2 \rfloor + 1$ bound pairs and hence $> n$ positions. This contradiction proves (iii).

iv) The argument used to prove the second part of (ii) suffices; namely for any chain C start with the minimal subset, P . This contains no free 1's. The next subset in the chain is formed by adding the element x_i to P where i is the rightmost (i.e maximal) free 0 in $\beta(P)$. i is now a free 1 in $\beta(P \cup \{x_i\})$ so the total number of free positions is

unchanged.

v) Using (iv) it need only be shown that $\beta(P_1 \cup (P_3 - P_2))$ contains 2 fewer free positions than $\beta(P_1)$. From (ii)

$$P_2 = P_1 \cup \{x_i\} ; P_3 = P_1 \cup \{x_i, x_j\}$$

for some $x_i, x_j \notin P_1$. Thus

$$P_1 \cup (P_3 - P_2) = P_1 \cup \{x_j\}$$

Both i and j must be free in $\beta(P_r)$, for all $1 \leq r \leq 3$. Since i is a free 1 in $\beta(P_2)$ so from Fact(3.1)(i) there are no free 0's at $j > i$. Thus $i > j$. Additionally since j is a free 1 in $\beta(P_3)$ there are no free 0's k such that $j < k < i$. It follows that the substring \underline{z} between positions $j+1$ and $i-1$ in $\beta(P_1)$ must be well-formed, for from Fact(3.1)(ii) there are no free 1's, k , such that $j+1 \leq k \leq i-1$. Now (v) follows easily since (j, i) is a bound pair in $\beta(P_1 \cup \{x_j\})$, thus this contains precisely 2 fewer free positions as required.

vi) With (iv) it is sufficient to show that the number of binary words of length n containing $n - 2r$ free 1's and no free 0's is exactly:

$$\binom{n}{r} - \binom{n}{r-1}$$

Any such word has $n - r$ 1's and r 0's. The total number of words with this many 0's and 1's is obviously $\binom{n}{r}$. However this includes words with fewer than r bound pairs. Let B denote the set of words with $n - r$ 1's, r 0's and at most $r - 1$ bound pairs. We shall show that the size of B is exactly $\binom{n}{r-1}$ by exhibiting a bijective mapping from B to the set of words containing exactly $n - r + 1$ 0's and $r - 1$ 1's. Subtracting this total from $\binom{n}{r}$ proves the desired bound.

We claim that any $\underline{w} = w_1 w_2 \cdots w_n$ with $n - r$ 1's and r 0's is in B if and only if there exists some i such that the subword $w_1 \cdots w_i$ contains more 0's than 1's.

The "if" part is easily verified since the first i at which this condition holds must be a free 0 and hence \underline{w} has fewer than r bound pairs. To establish "only if" consider some word \underline{w} in B . \underline{w} must contain a free 0 since it has r 0's but no more than $r - 1$ bound pairs. Let i be the lowest indexed free 0. $w_1 \cdots w_{i-1}$ is well-formed, from Fact(3.1)(ii) and the choice of i , and so contains equal numbers of 1's and 0's, thus $w_1 \cdots w_i$ contains precisely one more 0 than 1's.

From the previous paragraph it is easy to check the correctness of the following procedure which maps words in B to words with $n - r + 1$ 0's and $r - 1$ 1's, and vice-versa.

Input: \underline{w} in B (resp. \underline{z} containing $n - r + 1$ 0's and $r - 1$ 1's)

Output: \underline{z} containing $n - r + 1$ 0's and $r - 1$ 1's (resp. \underline{w} in B)

- 1) Find lowest indexed j such that $w_1 \cdots w_j$
(resp. $z_1 \cdots z_j$) has more 0's than 1's.
- 2) Complement the subword $w_{j+1} \cdots w_n$
(resp. $z_{j+1} \cdots z_n$).
i.e Change 1's to 0's, 0's to 1's

This completes the proof of (vi) and Fact(3.2). \square

Theorem 3.2: (Hansel, 1966) $\psi(n) \leq 3^n$

Proof: Consider the partition of $2^{\mathbf{X}_n}$ into E_n chains:

$$\{C_1, C_2, \dots, C_{E_n}\}$$

given in Fact(3.2). Let these be ordered by length so that $|C_i| \leq |C_{i+1}|$, $\forall 1 \leq i < E_n$. The ordering of chains of the same length is not important. Let the subsets of \mathbf{X}_n in the i 'th chain be denoted by:

$$\{P_i^1, P_i^2, \dots, P_i^{l_i}\}$$

where these are ordered by containment.

The upper bound is proved in two stages. In the first we show that any $f \in M_n$ may be encoded by a E_n ternary digit code,

$$t_1 t_2 \cdots t_i \cdots t_{E_n} \in \{0, 1, 2\}^{E_n}$$

In the second part it is proved that distinct functions map to *different* codes.

Observe that any function, $f \in M_n$ takes at most one monom from each chain as a prime implicant, since the chains are ordered by containment. We can thus encode any monotone Boolean function over \mathbf{X}_n by indicating which (if any) monom in each chain is a prime implicant of f . We shall prove by induction on the length of chains set so far, that there are at most 3 choices for the i 'th chain: namely no monom in C_i is a prime implicant of $f(\mathbf{X}_n)$ or P_i^j is or P_i^{j+1} is. (These possibilities will correspond to $t_i = 0, 2, 1$ respectively in the generated code). The key factor is that the exact position j within the i 'th chain does not need to be encoded; it can be deduced from the encoding of the previous $i-1$ chains. Note that from Fact(3.2)(vi) there are no even length chains if n is even; no odd length chains if n is odd. It is clear that for each chain of length ≤ 2 there are at most 3 choices as indicated above. So the inductive base is established. Now assume that for all chains of length $\leq n-2r-1$ (for some $0 \leq r < \lfloor n/2 \rfloor$) there have been at most 3 choices for each chain. We show that the same holds for each chain of length $n-2r+1$, namely that all but two monoms are predetermined as non-implicants or as (non-prime) implicants. Consider any chain, C_i , of length $n-2r+1$ and assume that a valid $t \in \{0, 1, 2\}$ has been assigned for each chain of length at most $n-2r-1$. Obviously as soon as one monom, P_i^k , is

selected as a prime implicant this precludes any monom $\supset P_i^k$ in the same chain being chosen. Now consider any consecutive sequence of subsets $P_i^j, P_i^{j+1}, P_i^{j+2}$ in C_i . From Fact(3.2)(v) the set $P_i^j \cup (P_i^{j+2} - P_i^{j+1})$ is in a chain of length $n - 2r - 1$ and so the corresponding monom, q say, is either not an implicant, in which event the monom defined by P_i^j is not an implicant (as $P_i^j \subset Q$), or q is an implicant, in which case the monoms defined by $P_i^k \forall k \geq j+2$ are also implicants since $Q \subset P_i^{j+2}$. The latter case leaves only 2 uncharacterised subsets, namely P_i^j, P_i^{j+1} . By considering each j from 1 up to $n - 2r + 1$ in this way, eventually at most 2 non-determined monoms remain. This completes the proof of the Inductive hypothesis.

It remains to show that distinct functions give rise to different codes. Let f and g be distinct functions in M_n . Since $f \neq g$ we have $\mathbf{PI}(f) \neq \mathbf{PI}(g)$. For each i $1 \leq i \leq E_n$ let:

$$p_i = \mathbf{PI}(f) \cap C_i \ ; \ q_i = \mathbf{PI}(g) \cap C_i$$

Let C_i be the lowest indexed chain for which $p_i \neq q_i$.

$$C_i = \{r_1, r_2, \dots, r_{j-2}, r_{j-1}, r_j = p_i, r_{j+1}, r_{j+2}, \dots\}$$

Since p_i is a prime implicant of f , but not of g and since the previous $i - 1$ chains have been encoded identically for f and g (by the choice of i) it follows that r_k is not an implicant of f or g , $\forall 1 \leq k \leq j - 3$. We distinguish 3 cases.

Case 1: r_j is determined as an implicant of both f and g , using Fact(3.2)(v) as in the first part of the proof. Then since $r_j = p_i$ is a *prime* implicant of f , neither r_{j-1} or r_{j-2} is an implicant of f , thus t_i must be 0 for f . However one of these must be a prime implicant of g , hence t_i is 1 or 2 for g .

Case 2: r_{j+1} is determined as an implicant of both f and g , as before. Using the same reasoning as Case(1), t_i must equal 1 for f , but 0 or 2 for g .

Case 3: r_{j+2} is determined as an implicant of both f and g . Note that this always holds since

$$R_j \subset R_j \cup (R_{j+2} - R_{j+1})$$

Again t_i must equal 2 for f , but 0 or 1 for g .

Thus in all 3 possibilities the code digit, t_i , assigned for C_i is different for f and g and this completes the proof of the second stage.

Now Theorem(3.2) follows easily. Since every function in M_n is specified by some n -digit ternary sequence \underline{t} , and there are exactly 3^n such sequences, we have $\psi(n) \leq 3^n$ as claimed. \square

Corollary 3.1: There exists a surjective mapping $CODE: \{0, 1, 2\}^{E_n} \rightarrow M_n$ \square

So given any E_n digit ternary code we can find a unique n -input monotone Boolean function associated with it and furthermore every such function is associated with some such code. Below we give a procedure which realises a surjective mapping from $\{0, 1, 2\}^{E_n}$ onto M_n . This determines the value of f over each chain, using the values of f which have already been determined and the ternary digit t_i . In order to test if the value of $f(\beta(P_i^j))$ is predetermined we employ two predicates: 0-covered takes a subset P of \mathbf{X}_n and the index, i , of some chain as parameters and is true if and only if $P \subset Q$ and $f(\beta(Q))$ has already been fixed to 0. Thus:

$$0 - Covered(P, i) \iff (P \subset Q \in \bigcup_{k=1}^i C_k \text{ and } f(\beta(Q)) = 0)$$

Similarly a subset P is 1-covering if and only if P is a superset of some subset Q such that $f(\beta(Q))$ has already been fixed to 1.

$$1 - \text{Covering}(P, i) \iff (P \supset Q \in \bigcup_{k=1}^i C_k \text{ and } f(\beta(Q)) = 1)$$

A procedure *Set Point*(i, j) is used.

```

proc Set Point( Chain:  $i$ , Chain element:  $j$ )
  if 0 - Covered( $P_i^j, i - 1$ ) then
     $f(\beta(P_i^j)) := 0$ 
  elif 1 - Covering( $P_i^j, i - 1$ ) then
     $f(\beta(P_i^j)) := 1$ 
  else  $f(\beta(P_i^j)) := \lfloor t_i/2 \rfloor$  fi
  if 0 - Covered( $P_i^{j+1}, i - 1$ ) then
     $f(\beta(P_i^{j+1})) := 1$ 
  elif 1 - Covering( $P_i^{j+1}, i - 1$ ) then
     $f(\beta(P_i^{j+1})) := 1$ 
  else  $f(\beta(P_i^{j+1})) := \lceil t_i/2 \rceil$  fi
corp

```

Input: $t_1 t_2 \cdots t_{E_n} \in \{0, 1, 2\}^{E_n}$

Output: $f(0, 0, \dots, 0), \dots, f(1, 1, \dots, 1)$

the truth-table of some $f \in M_n$

```

for each chain  $C_i$   $i = 1, 2, \dots, n$  do
  if  $|C_i| = 1$  then
     $f(\beta(P_i^1)) := \lfloor t_i/2 \rfloor$ 
  fi
  if  $|C_i| = 2$  then
    Set Point( $i, 1$ )
  fi
  if  $|C_i| \geq 3$  then
     $j := 1$ 
    (A) if 0-covered( $P_i^j, i-1$ ) then
       $f(\beta(P_i^j)) := 0; j := j+1$ 
    elif 1-covering( $P_i^j, i-1$ ) then
       $f(\beta(P_i^k)) := 1$  for  $k = j, j+1, \dots, i_r$ 
       $j := i_r$ 
    else {assert:  $f(\beta(P_i^j \cup (P_i^{j+2} - P_i^{j+1}))) = 1$ }
       $f(\beta(P_i^k)) := 1$  for  $k = j+2, j+3, \dots, i_r$ 
      Set Point( $i, j$ )
       $j := i_r$ 
    fi
    if  $j = i_r - 1$  then
      Set Point( $i, i_r - 1$ )
       $j := i_r$ 
    fi
    goto (A) if  $j \neq i_r$ 
  fi
od
    
```

We leave it to the reader to verify the correctness of this procedure.

The fact that such an encoding of all functions in M_n exists and can be easily applied turns out to be of great value in constructing efficient *combinational* networks for any monotone Boolean function. We shall only briefly outline how this may be done using Hansel's procedure above. For a fuller description the reader should consult Pippenger (1978). The technique applied derives from the ideas of Lupanov (1961b, 1965b) and is known as "The Principle of Local Coding".

Consider any class $H_n \subseteq B_n$ of Boolean functions, with the property that for all functions $h \in H_n$ every subfunction of h is in $\bigcup_{i=0}^{n-1} H_i$. The class M_n is a particular example. We have seen earlier how a lower bound on $|H_n|$ may be used to obtain lower bounds on the combinational complexity of almost all functions in H_n . The local coding principle is a method of deriving, under suitable conditions, upper bounds on the combinational complexity of any function $h \in H_n$. Suppose that for each n , there exists a surjective mapping from $\{0, 1\}^p \rightarrow H_n$ so that every function in H_n can be associated with a distinct p -digit binary codeword $C_1 \cdots C_p$. Of course the precise value of p will depend on n , to keep the notation as simple as possible we shall not make this explicit unless required. If these conditions are all met then we can employ a network of the form in Figure(3.1) to compute any $h \in H_n$. This network consists of 3 parts which fulfill the following functions. The inputs, \mathbf{X}_n , are partitioned into two sets $\{x_1, \dots, x_m\}$ and $\{x_{m+1}, \dots, x_n\}$. Any subfunction of h induced by an assignment to $\{x_{m+1}, \dots, x_n\}$ belongs to H_m and thus can be associated with some unique p digit binary codeword. The network N'' of Fig(3.1) outputs the binary codeword $C_1 \cdots C_p$

Figure 3.1

corresponding to the subfunction of h arising from a given assignment to $\langle x_{m+1} \cdots x_n \rangle$. Note that this network just computes some set of functions in $B_{n-m,p}$. The codeword produced is fed as input to the second stage, U . This is a decoding network, having p inputs and 2^m outputs, these outputs being the truth-table of the function in H_m encoded by $C_1 \cdots C_p$. The final stage, N' , is a selector network: the output of U which corresponds to the assignment in $\{0,1\}^m$ given to $x_1 \cdots x_m$ is chosen by N' and returned as the result of h . Upper bounds on the combinational complexity of N' and N'' may be

obtained (relatively) easily. The problem in applying the construction in general is that of finding an efficient encoding scheme and decoding network.^{d)}

The following lemmas give upper bounds on the combinational complexity of N' and N'' .

Lemma 3.1: $\forall n$ there exists an $n + 2^n$ input, single output network which computes the value of h from the truth-table of h and \mathbf{X}_n which network contains $O(2^n)$ gates.

Proof: See Lupanov (1965b), Lemma(2.2), (p.42). \square

Lemma 3.2: Let $C(B_{n,m})$ denote the maximal combinational complexity of any n -input, m -output function in $B_{n,m}$. If $3 \log \log m \leq n + O(1)$ then,

$$C(B_{n,m}) \leq \frac{m2^n}{n + \log m} \exp\left(O\left(\frac{\log \log [m2^n]}{n + \log m}\right)\right)$$

Proof: See Lupanov (1965b), Theorem(D.13), (p.109). \square

Hansel's result shows that any $f \in M_n$ can be encoded by a binary codeword of length $p(n) = \lceil (\log 3) E_n \rceil$. Using Hansel's procedure above, an efficient decoding network can be built.

Lemma 3.3: Let U_n be a minimal combinational network which produces the truth-table of a function in M_n from its $p(n)$ -digit codeword.

$$C(U_n) = O(E_n (\log E_n)^r)$$

where r is some constant.

d) The universal construction for symmetric functions given in Chapter(2) is a particularly simple application of this principle in which the Decoder and Selector are collapsed into a single network. The codeword consists of $\lceil \log(n+1) \rceil$ digits being the binary representation of the number of 1's in the assignment to \mathbf{X}_n . The Decode/Select stage then just returns the result using the function spectrum.

Proof: A detailed construction is given in Pippenger (1978). \square

Theorem 3.3: (Hansel, 1966) $\forall f \in M_n, \forall \varepsilon > 0$ and sufficiently large n :

$$C(f) < (\log 3) \sqrt{\frac{2}{\pi}} \frac{(1 + \varepsilon)2^n}{n^{3/2}}$$

Proof: We apply the principle of local coding and the results of the three preceding lemmas. Let the inputs \mathbf{X}_n be divided into $\{x_1, \dots, x_m\}$ and $\{x_{m+1}, \dots, x_n\}$ and compute f using the scheme of Figure(3.1) and the decoding network U whose existence is established by Lemma(3.3). We then have:

N'' is a network with $n - m$ inputs and $\lceil (\log 3)E_m \rceil$ outputs. For suitable choice of m , its complexity will be:

$$\frac{\lceil (\log 3)E_m \rceil 2^{n-m}}{n - m + \log(\lceil (\log 3)E_m \rceil)}$$

U is a network with $\lceil (\log 3)E_m \rceil$ inputs and 2^m outputs. From Lemma(3.3) its complexity is:

$$O(E_m(\log E_m)^r)$$

N' is a network with $m + 2^m$ inputs and a single output. It has complexity $O(2^m)$.

m must be chosen so that:

$$3 \log \log(\lceil (\log 3)E_m \rceil) \leq n - m + O(1)$$

Choosing $m = n - c \log n$ for some constant c depending on r ensures this. It may now be easily verified that the complexity of N'' is the dominating term, and that with the choice of m this yields:

$$C(f) \leq (\log 3) \sqrt{\frac{2}{\pi}} \frac{(1 + \varepsilon)2^n}{n^{3/2}}$$

for all $\varepsilon > 0$ and n large enough. \square

Pippenger (1976, 1978) and independently Ugolnikov (1976), prove that the lower bound $\sqrt{\frac{2}{\pi}} \frac{(1 - \varepsilon)2^n}{n^{3/2}}$ is the best possible for combinational complexity, by obtaining bounds of:

$$C(M_n) \sim \sqrt{\frac{2}{\pi}} \frac{2^n}{n^{3/2}} \quad (\text{Ugolnikov, 1976})$$

$$C(M_n) = \sqrt{\frac{2}{\pi}} \frac{2^n}{n^{3/2}} \left(1 + O\left(\frac{\log n}{n}\right) \right) \quad (\text{Pippenger, 1976, 1978})$$

The latter result also utilises the principle of local coding in conjunction with the construction of Kleitman and Markowsky (1974).

A more natural question for monotone computation concerns the monotone network complexity of monotone Boolean functions. Namely, to determine upper and lower bounds on $\mathbf{C}^m(M_n) = \max \{f \in M_n : \mathbf{C}^m(f)\}$. A lower bound is again easily obtained from Shannon's methods. Red'kin (1979) gives a construction which asymptotically matches this, improving the earlier $\frac{2^n \log n}{n^{3/2}}$ bound of Pippenger (1976). This again relies heavily on the partition of $2^{\mathbf{X}_n}$ into E_n connected, symmetric chains described in Fact(3.2).

Theorem 3.4: (Red'kin 1979) $\forall f \in M_n$

$$\mathbf{C}^m(f) = O\left(\frac{2^n}{n^{3/2}}\right)$$

Before proving this theorem we need some preliminary results. Let \mathbf{X}_n be partitioned into two sets \mathbf{Y} and \mathbf{Z} of sizes n_1 and n_2 . Further, for any set of Boolean variables, \mathbf{W} , let $\Pi_{\mathbf{W}}$ denote the set of chains constituting the partition of $2^{\mathbf{W}}$ discussed in Fact(3.2).

Given $f \in M_n$ and chains $Q_k \in \Pi_{\mathbf{Y}}$, $R_l \in \Pi_{\mathbf{Z}}$ having length k and l respectively, we define the function $f_{Q_k \times R_l}(\mathbf{Y}, \mathbf{Z})$ by,

$$f_{Q_k \times R_l}(\pi, \sigma) = \begin{cases} f(\pi, \sigma) & \text{if } \pi \in Q_k \text{ and } \sigma \in R_l \\ 0 & \text{otherwise} \end{cases}$$

In this $\pi \in \{0, 1\}^{n_1}$ and $\pi \in Q_k$ is a shorthand for $\beta^{-1}(\pi) \in Q_k$, and similarly for $\sigma \in \{0, 1\}^{n_2}$. β is the mapping from monoms over \mathbf{Y} to binary words described after Fact(3.1). To avoid confusion when distinguishing different sets of variables subsequently we will use $\gamma_{\rho}(\mathbf{W})$ to denote the monom over \mathbf{W} corresponding to the binary word ρ . i.e

$$\gamma_{\rho}(\mathbf{W}) = \bigvee_{w \in \beta^{-1}(\rho)} w$$

In general this function will not be monotone, however we can define a monotone "approximation" to it which will be sufficient for Thm(3.4).

Thus, $f_{Q_k \times R_l}^+(\mathbf{Y}, \mathbf{Z})$ is given by:

$$\bigvee_{\{(\pi, \sigma) \in Q_k \times R_l : f_{Q_k \times R_l}(\pi, \sigma) = 1\}} \gamma_{\pi}(\mathbf{Y}) \gamma_{\sigma}(\mathbf{Z})$$

Now for $0 \leq p \leq \lfloor n/2 \rfloor$ let $h(n, p) = \binom{n}{p} - \binom{n}{p-1}$ and recall that there are exactly $h(n, p)$ chains of length $n - 2p + 1$ in $\Pi_{\mathbf{X}_n}$. Clearly for any $f \in M_n$ we have,

$$f(\mathbf{Y}, \mathbf{Z}) = \prod_{p=0}^{\lfloor n_1/2 \rfloor} \prod_{i=1}^{h(n_1,p)} \prod_{q=0}^{\lfloor n_2/2 \rfloor} \prod_{j=1}^{h(n_2,q)} f_{p,i,q,j}^+(\mathbf{Y}, \mathbf{Z}) \quad (3.1)$$

where $f_{p,i,q,j}^+(\mathbf{Y}, \mathbf{Z})$ is

$$f_{Q_i, \lfloor n_1/2 \rfloor - 2p + 1 \times R_j, \lfloor n_2/2 \rfloor - 2q + 1}^+(\mathbf{Y}, \mathbf{Z}) \quad (3.2)$$

$Q_{i,r}$ being the i 'th chain of length r under some ordering.

The expansion defined by (3.1) and (3.2) is central to Red'kin's upper bound construction. Before presenting this we require one preliminary result.

Fact 3.3: Let $Q_k \in \Pi_{\mathbf{Y}}$, $R_l \in \Pi_{\mathbf{Z}}$ be chains of length k and l as before. Then

$$\begin{aligned} |\{f_{Q_k \times R_l}^+ : f \in M_n\}| &\leq |\{f_{Q_k \times R_l} : f \in M_n\}| \\ &= \binom{k+l}{k} \end{aligned}$$

Proof: The first inequality is immediate from the definitions of f_{\dots}^+ and f_{\dots} . For the second observe that we can represent $f_{Q_k \times R_l}$ as a $k \times l$ Boolean matrix, M , in which $M_{i,j}$ is the value of $f(\pi_i, \sigma_j)$; π_i being the i 'th element of the ordered chain Q_k , σ_j the j 'th element of the ordered chain R_l . It follows that it is sufficient to count the number of pairwise distinct matrices, M , which are consistent with f being monotone.

Since,

$$\begin{aligned} Q_k &= \langle \pi_1, \pi_2, \dots, \pi_k \rangle \\ R_l &= \langle \sigma_1, \sigma_2, \dots, \sigma_l \rangle \end{aligned}$$

with $\gamma_{\pi_{i+1}}(\mathbf{Y}) < \gamma_{\pi_i}(\mathbf{Y})$, $\gamma_{\sigma_{j+1}}(\mathbf{Z}) < \gamma_{\sigma_j}(\mathbf{Z})$ for each $1 \leq i < k$, $1 \leq j < l$, it follows

$$f(\pi_i, \sigma_j) = 1 \Leftrightarrow f(\pi_{i+s}, \sigma_{j+t}) = 1$$

for each valid $s \geq 0$, $t \geq 0$. So in counting the number of distinct appropriate matrices we know that for each row i , if $M_{i,j} = 1$ then the values of $M_{i,j+t}$ are predetermined to be 1 also. We can now proceed with an inductive argument. The result is obvious for $k = 1$ so assume it holds for all values $\leq k - 1$ and all l and consider the number of valid $k \times l$ matrices. By the inductive hypothesis, there are exactly $\binom{k-1+l}{k-1}$ valid matrices in which the first row is entirely 0. Similarly for each $1 \leq j \leq l$ there are exactly $\binom{k-1+j-1}{k-1}$ matrices in which the first $j-1$ entries of the first row are 0's and the remainder 1's. Since there are no other consistent assignments to the first row we have that the total number of valid $k \times l$ matrices is exactly

$$\sum_{j=1}^{l+1} \binom{k-1+j-1}{k-1}$$

and an easy induction on $l \geq 1$ shows this to be $\binom{k+l}{l}$ as claimed. \square

Proof of Theorem 3.4: Let $f \in M_n$ and partition \mathbf{X}_n into 3 disjoint sets of variables: $\mathbf{W} = \langle x_1, \dots, x_{n-2m} \rangle$; $\mathbf{Y} = \langle x_{n-2m+1}, \dots, x_{n-m} \rangle$; and $\mathbf{Z} = \langle x_{n-m+1}, \dots, x_n \rangle$ where m will be fixed subsequently. Since f is monotone it is clear that

$$f(\mathbf{W}, \mathbf{Y}, \mathbf{Z}) = \bigvee_{\alpha \in \{0,1\}^{n-2m}} \gamma_{\alpha}(\mathbf{W}) f^{\mathbf{W}:=\alpha}(\mathbf{Y}, \mathbf{Z}) \quad (3.3)$$

Using r to denote $\lfloor m/2 \rfloor$ (3.1) and (3.3) show that $f(\mathbf{W}, \mathbf{Y}, \mathbf{Z})$ is

$$\bigvee_{p=0}^r \bigvee_{i=1}^{h(m,p)} \bigvee_{q=0}^r \bigvee_{\alpha} \bigvee_{j=1}^{h(m,q)} \gamma_{\alpha}(\mathbf{W}) f_{p,i,q,j}^{\gamma_{\alpha}}(\mathbf{Y}, \mathbf{Z}) \quad (3.4)$$

Hence,

$$\mathbf{C}^m(f) \leq \sum_{p=0}^r \sum_{i=1}^{h(m,p)} \sum_{q=0}^r \mathbf{C}^m(f_{p,i,q}(\mathbf{W}, \mathbf{Y}, \mathbf{Z})) \quad (3.5)$$

$f_{p,i,q}$ being the inner two \vee levels of (3.4). We group the chains of length $m - 2q + 1$ into blocks of size at most $s(p, q)$. $s(p, q)$ is chosen so that

$$\binom{2(m-p-q+1)}{m-2q+1}^{s(p,q)} \leq 2^{n-3m} < \binom{2(m-p-q+1)}{m-2q+1}^{s(p,q)+1} \quad (3.6)$$

Now for $n \geq 8$ and $m = \lfloor (n-2)/6 \rfloor$ it holds,

$$2 \leq \binom{2(m-p-q+1)}{m-2q+1} \leq 2^{n-3m}$$

From the first inequality and (3.6) we deduce that $s(p, q) \leq n - 3m$.

From (3.6) and the fact that $\binom{n}{k} \leq 2^n$ it follows that

$s(p, q) > \frac{n-3m}{2(m-p-q+1)} - 1$. In total

$$\frac{n-3m}{2(m-p-q+1)} - 1 < s(p, q) \leq n - 3m \quad (3.7)$$

So, realising (3.5) by grouping chains of length $m - 2q + 1$ into sets of size $s(p, q)$ gives $f_{p,i,q}(\mathbf{W}, \mathbf{Y}, \mathbf{Z})$ as

$$\bigvee_{\alpha} \gamma_{\alpha}(\mathbf{W}) \left[\begin{array}{c} h(m,q)/s(p,q) \\ \bigvee \\ t=1 \end{array} \right] f_{p,i,q,t}^{\alpha}(\mathbf{Y}, \mathbf{Z}) \quad (3.8)$$

where

$$f_{p,i,q,t}^{\alpha} = \min\{s(p,q)t, h(m,q)\} \bigvee_{j=s(p,q)(t-1)+1} f_{p,i,q,j}^{\alpha+}(\mathbf{Y}, \mathbf{Z})$$

Hence,

$$\mathbf{C}^{\mathbf{m}}(f_{p,i,q}) \leq \sum_{t=1}^{\lceil h(m,q)/s(p,q) \rceil} \mathbf{C}^{\mathbf{m}}(f_{p,i,q,t}^+) \quad (3.9)$$

where

$$f_{p,i,q,t}^+ = \bigvee_{\alpha} \gamma_{\alpha}(\mathbf{W}) f_{p,i,q,t}^{\alpha}(\mathbf{Y}, \mathbf{Z})$$

We can now describe a 4 part monotone network, S , realising $f_{p,i,q,t}^+$ for fixed p, i, q and t . S consists of sub-networks S_1, S_2, S_3 and S_4 . S_1 has inputs \mathbf{W} and realises all the monoms over \mathbf{W} , i.e the functions $\gamma_{\alpha}(\mathbf{W})$ for each α . Using the $n - 2m$ -ordered network \mathbf{U}_{n-2m} in which instances of \bar{x}_i are replaced by the constant function 1, this can be accomplished in at most 2^{n-2m} \wedge -gates.

The network S_2 has inputs $\mathbf{Y} \cup \mathbf{Z}$ and realises all of the functions $f_{p,i,q,t}^{\gamma}$. Each of these is the disjunction of at most $s(p, q)$ functions of the form,

$$f_{Q_{i,m-2p+1} \times R_{j,m-2q+1}}^{\alpha+} \quad (3.10)$$

For fixed i, p, j, q and the number of distinct functions over $\mathbf{Y} \cup \mathbf{Z}$ of the form of (3.10) is, from Fact(3.3), at most

$$g(m, p, q) = \left| \bigcup_{\alpha} \{ f_{Q_{i,m-2p+1} \times R_{j,m-2q+1}}^{\alpha+} \} \right|$$

$$\leq \binom{2(m-p-q+1)}{m-p-q+1}$$

So the number of distinct functions $f_{p,i,q,t}^\alpha$, for fixed p, i, q , and t is at most $g(m, p, q)^{s(p,q)}$ which from (3.6) does not exceed 2^{n-3m} .

Each of the functions from (3.10) has by definition at most $(m+1)^2$ prime implicants, each prime implicant containing at most $2m$ variables. Hence a single $f_{p,i,q,t}^\alpha$ can be realised in $s(m+1)^2 2m$ gates, and all of them in at most $2^{n-3m} s(m+1)^2 2m$ gates.

S_3 conjoins each output of S_1 to its appropriate output from S_2 and thus has at most 2^{n-2m} gates. Finally S_4 \vee 's together all the outputs of S_4 , adding a further 2^{n-2m} gates.

In total we have,

$$\mathbf{C}^m(f_{p,i,q,t}^+) \leq c_1 2^{n-2m} + 2^{n-3m} s(m+1)^2 2m$$

which is $\leq c_2 2^{n-2m}$ for some constant c_2 , given that $m = \lfloor (n-2)/6 \rfloor$.

Combining this with our previous expansions it follows that,

$$\begin{aligned} \mathbf{C}^m(f) &\leq \sum_{p=0}^r \sum_{i=1}^{h(m,p)} \sum_{q=0}^r \sum_{t=1}^{\lceil h(m,q)/s(p,q) \rceil} c_2 2^{n-2m} \\ &\leq c_2 2^{n-2m} \sum_{p=0}^r \sum_{i=1}^{h(m,p)} \sum_{q=0}^r \left(\frac{h(m,p)}{(n-3m)/2(m-p-q+1)} + 1 \right) \\ &\leq c_3 \frac{2^{n-2m}}{n} \sum_{p=0}^r \sum_{i=1}^{h(m,p)} \sum_{q=0}^r (2(m-p-q+1)h(m,q) + m) \end{aligned}$$

It is relatively straightforward to show that,

$$\sum_{i=0}^r (m - 2i + 2)h(m, i) = 2^m + E_m$$

and consequently,

$$\begin{aligned} \mathbf{C}^{\mathbf{m}}(f) &\leq c_3 \frac{2^{n-2m}}{n} \sum_{p=0}^r \sum_{i=1}^{h(m,p)} ((m - 2p + 2) E_m + 2^m + m^2) \\ &< c_4 \frac{2^{n-2m}}{n} \sum_{p=0}^r ((m - 2p + 2) h(m, p) E_m + 2^m E_m) \\ &= c_4 \frac{2^{n-2m}}{n} ((2^m + E_m) E_m + 2^m E_m) \\ &< c_5 \frac{2^n}{n^{3/2}} \end{aligned}$$

from the choice of m . This proves Theorem(3.4). \square

We conclude this section by presenting a very slight improvement to the complexity hierarchy for monotone network size. This divides the upper range $[\mathbf{C}^{\mathbf{m}}(M_{n-1}), \mathbf{C}^{\mathbf{m}}(M_n)]$ into two parts: $[\mathbf{C}^{\mathbf{m}}(M_{n-1}), \mathbf{C}^{\mathbf{m}}(P_n^s)]$ and $[\mathbf{C}^{\mathbf{m}}(P_n^s), \mathbf{C}^{\mathbf{m}}(M_n)]$. The class of monotone Boolean functions P_n^s was considered by Korshunov (1981). Its precise definition is not important, the only property of it that we require is:

For $s \in \{\lfloor n/2 \rfloor, \lceil n/2 \rceil\}$:

If n is even and $f \in P_n^s$ then all prime implicants of f contain between $\frac{n}{2} - 1$ and $\frac{n}{2} + 2$ variables

If n is odd and $f \in P_n^s$ then all prime implicants of f contain between $s - 1$ and $s + 2$ variables.

Fact 3.4: (Korshunov, 1981) Let $M_n^* = P_n^{n/2}$ if n is even and $M_n^* = P_n^{\lfloor n/2 \rfloor} \cup P_n^{\lceil n/2 \rceil}$ if n is odd. Almost all $f \in M_n$ are in M_n^* . \square

Fact 3.5: If $f \in M_n$, such that $\mathbf{C}^m(f) \geq \mathbf{C}^m(M_{n-1})$ then f has a prime implicant, which contains at most $n - \lceil \frac{n}{\log_2 n} \rceil$ variables.

Proof: This is an easy counting argument. \square

With these two facts the hierarchy result below, improving Thm(2.12), for monotone bases, is immediate.

Theorem 3.5:

$$c_{\{\wedge, \vee\}}(r) \leq \lceil \frac{n}{2} \rceil + 2 \quad \text{for } \mathbf{C}^m(M_{n-1}) \leq r < \mathbf{C}^m(P_n^s)$$

$$c_{\{\wedge, \vee\}}(r) \leq n - \lceil \frac{n}{\log_2 n} \rceil \quad \text{for } \mathbf{C}^m(P_n^s) \leq r < \mathbf{C}^m(M_n)$$

\square

3.2) Replacement Rules

One reason for the failure of inductive methods to derive super-linear lower bounds on combinational network size lies in the fact that such methods have made little use of the structure of optimal networks. For the unrestricted case information about the form of minimal networks appears to be very difficult to obtain. In contrast many inductive proofs concerned with the complexity of monotone computation rely on arguments which assert that optimal monotone networks, realising $f \in M_n$, do not contain gates computing certain functions. Such arguments utilise a powerful technique called *replacement rules*, which was independently proposed by Paterson (1975) and (Mehlhorn and Galil, 1976).

Definition 3.2: A replacement rule for $f \in M_n$ is a rule of the form:

In any monotone network S computing f , any node u , for which $res(u) = g$ may be replaced by a node w for which $res(w) = h$ and the resulting monotone network will still compute f . •

Here "replaced" means that the node u is deleted from S , together with any wires $\langle v, u \rangle$ or $\langle u, r \rangle$; then the node w is added to $S - u$ and wires $\langle w, r \rangle$ for all r such that $\langle u, r \rangle$ was a wire in S . We shall say that " g is h -replaceable with respect to f ", denoting this by $g \stackrel{f}{\equiv} h$. It is important to note that replacements are universally valid in the sense that g may be replaced by h in *all* monotone networks realising f .

How can this idea be useful in determining the structure of optimal monotone networks? Suppose that for some $f \in M_n$ we know that $g \stackrel{f}{\equiv} 0$, or $g \stackrel{f}{\equiv} 1$ or $g \stackrel{f}{\equiv} x_i$. Then no optimal monotone network realising f can contain a gate whose result is g , for any such gate can be eliminated and replaced by a constant function or an input x_i . For suitable f and g determination of such rules may allow significant properties of optimal networks to be inferred. For example Pater-son (1975) considers the monotone complexity of Boolean matrix product, $BMP(\mathbf{X}_{n,n}, \mathbf{Y}_{n,n}): \{0, 1\}^{2n^2} \rightarrow \{0, 1\}^{n^2}$ where each output c_{ij} is defined by

$$c_{ij} = \bigvee_{k=1}^n (x_{ik} \wedge y_{kj})$$

By showing that certain functions can be replaced by 1, it is deduced that there is one \wedge -gate for every prime implicant of each c_{ij} . It follows that $\mathbf{C}^m(BMP) \geq n^3$ since this number of \wedge -gates is required.

In this section we examine general replacement rules. First two results giving necessary and sufficient conditions for a functions to be replaceable by a constant functions are proved. These originated in (Mehlhorn and Galil, 1976). In the remainder of this section we briefly survey some characterisation results of (Dunne, 1984a,c) which yield closed form expressions describing all valid replacement rules.

Definition 3.3: Let p be a monom and c a clause defined over subsets of \mathbf{X}_n . The monotone Boolean functions $\chi(p)$ and $\phi(c)$ are defined by:

$$\chi(p) = \bigvee_{\{x_i \in \mathbf{X}_n - \text{var}(p)\}} x_i$$

$$\phi(c) = \bigwedge_{\{x_i \in \mathbf{X}_n - \text{var}(c)\}} x_i \quad \bullet$$

Lemma 3.4:

i) (Mehlhorn and Galil, 1976)

$$g \stackrel{f}{\equiv} 0 \iff \forall p \in \mathbf{PI}(g) \neg \exists m \text{ s.t } pm \in \mathbf{PI}(f)$$

ii) $g \stackrel{f}{\equiv} 0 \iff f \leq g \vee h \iff f \leq h$

iii) (Dunne, 1984a,c) $g \stackrel{f}{\equiv} 0 \iff 0 \leq g \leq \bigwedge_{p \in \mathbf{PI}(f)} \chi(p)$

Proof:

i) Since

$$g_1 \stackrel{f}{\equiv} 0 \text{ and } g_2 \stackrel{f}{\equiv} 0 \iff g_1 \vee g_2 \stackrel{f}{\equiv} 0$$

it is sufficient to prove (i) for g being a single monom, p say. So

suppose $p \stackrel{f}{\equiv} 0$ but that there exists a monom m for which $pm \in \mathbf{PI}(f)$. Thus

$$f = pm \vee \bigvee_{q \in \mathbf{PI}(f) - pm} q$$

but $\bigvee_{q \in \mathbf{PI}(f) - pm} q \not\equiv f$. This contradicts $p \stackrel{f}{\equiv} 0$ and so no such m can exist. On the other hand, suppose that there does not exist any monom m such that $pm \in \mathbf{PI}(f)$. Let S be any monotone network realising f at some node t and let u be a node in S for which $res(u) = p$. Consider any path from u to t ; this may be regarded as computing some function R of p and \mathbf{X}_n ;

$$R(p, \mathbf{X}_n) = p R_1 \vee R_2 \leq f$$

since S realises f . It follows that $p R_1 \leq f$ but $\mathbf{PI}(p R_1) \cap \mathbf{PI}(f) = \{\}$ and so replacing the node u by 0 cannot affect the computation of f at t .

ii) \Rightarrow Suppose $g \stackrel{f}{\equiv} 0$ and that $f \leq g \vee h$ for some h . We wish to show that $f \leq h$. Since $f \leq g \vee h$, so $f = (g \vee h) f$. But $g \stackrel{f}{\equiv} 0$ hence $f = h f$ thus $f \leq h$.

\Rightarrow As in (i) let S be a monotone network realising f at t and containing a node u with result g . Consider the function of g and \mathbf{X}_n computed on some path from u to t , i.e $g h_1 \vee h_2$. Since S computes f , we have,

$$f \leq g h_1 \vee h_2 = (g \vee h_2)(h_1 \vee h_2)$$

$$\leq g \vee h_2 \Rightarrow f \leq h_2$$

So replacing u by 0 does not affect the computation of f by S , i.e. $g \stackrel{f}{=} 0$.

iii) \Leftrightarrow Suppose that $g \stackrel{f}{=} 0$. From (i) for all prime implicants, q , of g there does not exist any m for which $qm \in \mathbf{PI}(f)$. Thus for every prime implicant, p , of f

$$p \not\leq q \Leftrightarrow \exists x_i \in \text{var}(q) - \text{var}(p)$$

$$\Leftrightarrow q \leq \chi(p)$$

So $q \leq \bigwedge_{p \in \mathbf{PI}(f)} \chi(p)$ for every $q \in \mathbf{PI}(g)$

\Rightarrow Suppose that $0 \leq g \leq \bigwedge_{p \in \mathbf{PI}(f)} \chi(p)$. Consider any $q \in \mathbf{PI}(g)$ and any $p \in \mathbf{PI}(f)$. By the choice of g , $q \leq \chi(p)$ thus $p \not\leq q$ and so there does not exist any monom m for which $qm = p$. Since p and q were chosen arbitrarily, it follows from (i) that $g \stackrel{f}{=} 0$. \square

Lemma 3.5:

$$\text{i) } g \stackrel{f}{=} 1 \Leftrightarrow$$

$$\forall c \in \mathbf{PC}(g) \neg \exists s \text{ such that } c \vee s \in \mathbf{PC}(f)$$

$$\text{ii) } g \stackrel{f}{=} 1 \Leftrightarrow g \wedge h \leq f \Leftrightarrow h \leq f$$

$$\text{iii) } g \stackrel{f}{=} 1 \Leftrightarrow \bigvee_{c \in \mathbf{PC}(f)} \phi(c) \leq g \leq 1$$

Proof: Duality. \square

The next theorem completely characterises all valid replacement rules in monotone Boolean networks.

Definition 3.4: Let $M = \{m_1, \dots, m_k\}$ be a set of monoms, and let f be a monotone Boolean function. The *Prime-Implicant Extension* of M with respect to f ($\mathbf{IE}_f(M)$) is defined as,

$$\mathbf{IE}_f(M) = \{ p \in \mathbf{PI}(f) \mid \exists m_i \in M \text{ with } p \leq m_i \}$$

The *Prime-Clause Extension* of a set of clauses $C = \{c_1, \dots, c_k\}$ with respect to f ($\mathbf{CE}_f(C)$) is given by,

$$\mathbf{CE}_f(C) = \{ p \in \mathbf{PC}(f) \mid \exists c_i \in C \text{ with } c_i \leq p \}$$

In addition let,

$$A(f, g) = \bigvee_{m \in \mathbf{IE}_f(\mathbf{PI}(g))} m$$

$$B(f, g) = \bigwedge_{c \in \mathbf{CE}_f(\mathbf{PC}(g))} c$$

Note: Conventionally the empty monom (clause) is 1 (0).

$$\mathbf{PI}_{\text{rem}}(f, g) = \mathbf{PI}(f) - \mathbf{IE}_f(\mathbf{PI}(g))$$

$$\mathbf{PC}_{\text{rem}}(f, g) = \mathbf{PC}(f) - \mathbf{CE}_f(\mathbf{PC}(g))$$

$$E(f, g) = \bigwedge_{m \in \mathbf{PI}_{\text{rem}}(f, g)} \chi(m)$$

$$D(f, g) = \bigvee_{c \in \mathbf{PC}_{\text{rem}}(f, g)} \phi(c) \quad \bullet$$

Theorem 3.6: $g \stackrel{f}{\Longrightarrow} h \iff$

$$\text{R1) } A(f, g) \leq h \leq B(f, g)$$

$$\text{R2) } D(f, h) \leq g \leq E(f, h)$$

Proof:

R1) \Rightarrow is obvious.

\Leftarrow Since $\tilde{A}(f, g) = B(\tilde{f}, \tilde{g})$ we have

$$A(f, g) \leq h \leq B(f, g) \iff A(\tilde{f}, \tilde{g}) \leq \tilde{h} \leq B(\tilde{f}, \tilde{g})$$

Thus to prove \Leftarrow it is sufficient to show that $A(f, g) \leq h$ and then $h \leq B(f, g)$ follows by duality. Suppose the contrary and that $A(f, g) \not\leq h$. There exists some $p \in \mathbf{PI}(A(f, g))$ such that $p \not\leq h$. It will be shown that this contradicts $g \stackrel{f}{=} h$. By the definition of $A(f, g)$ we have $p \leq g$ and p is a prime implicant of f , so as in Lemma(3.4)(i),

$$f = pg \vee \bigvee_{q \in \mathbf{PI}(f)-p} q$$

but $f \not\leq ph \vee \bigvee_{q \in \mathbf{PI}(f)-p} q$.

R2) \Leftarrow Again since $\tilde{E}(f, h) = D(\tilde{f}, \tilde{h})$ it suffices to prove that $g \stackrel{f}{=} h \Leftarrow g \leq E(f, h)$, for then $D(f, h) \leq g$ follows by duality. Suppose that $g \not\leq E(f, h)$, so that there is some $p \in \mathbf{PI}(g)$ for which $p \not\leq E(f, h)$. Thus $E(f, h) \leq \chi(p)$ and from the definition of E there is some $r \in \mathbf{PI}_{\text{rem}}(f, h)$ such that

$$E(f, h) \leq \chi(r) \leq \chi(p)$$

so $r \leq p$. Now r is a prime implicant of f so

$$f = gr \vee \bigvee_{q \in \mathbf{PI}(f)-r} q$$

but

$$f \not\leq hr \vee \bigvee_{q \in \mathbf{PI}(f)-r} q$$

since $r \in \mathbf{PI}_{\text{rem}}(f, h)$ and this contradicts $g \stackrel{f}{\equiv} h$.

⇒ Clearly

$$\mathbf{IE}_f(g) \subseteq \mathbf{IE}_f(h) \wedge \mathbf{CE}_f(g) \subseteq \mathbf{CE}_f(h) \Leftrightarrow g \stackrel{f}{\equiv} h$$

So it is sufficient to prove that

$$\begin{aligned} g \leq E(f, h) &\Leftrightarrow \mathbf{IE}_f(g) \subseteq \mathbf{IE}_f(h) \\ D(f, h) \leq g &\Leftrightarrow \mathbf{CE}_f(g) \subseteq \mathbf{CE}_f(h) \end{aligned}$$

The latter following, by duality, from the former. Suppose, then, that $g \leq E(f, h)$ but $\mathbf{IE}_f(g) \not\subseteq \mathbf{IE}_f(h)$. In this case there exists some $p \in \mathbf{IE}_f(g)$ such that $p \notin \mathbf{IE}_f(h)$, thus $p \in \mathbf{PI}_{\text{rem}}(f, h)$. Now $p \not\leq \chi(p)$ hence

$$p \not\leq \bigwedge_{q \in \mathbf{PI}_{\text{rem}}(f, h)} q = E(f, h)$$

This contradiction proves $\mathbf{IE}_f(g) \subseteq \mathbf{IE}_f(h)$. □

3.3) The Monotone Complexity of Sets of Functions

Replacement rules in combination with inductive gate elimination provides a technique sufficiently powerful to obtain superlinear^{e)} lower bounds on the monotone complexity of several multiple output functions. The form such arguments take consists of identifying classes of functions which may be replaced by constants, and

e) i.e $\omega(n + m)$ for functions in $M_{n,m}$.

therefore cannot occur as intermediate results in minimal networks. For suitable classes the means by which optimal networks realise the output functions will be severely constrained. In this way the knowledge of the structure of minimal networks may permit many gates to be eliminated in the inductive step.

In this section the approach described above is illustrated with three examples: The results of Paterson (1975) concerning the monotone complexity of $\{\wedge, \vee\}$ -Boolean matrix product; the lower bound of Weiss (1983) on the number of \vee -gates required to compute Boolean Convolution; and the result of Mehlhorn (1979), which proves lower bounds for realising certain sets of Boolean sums.

The following notion is used in some proofs below.

Definition 3.5: Let Π be some predicate defined on the gates of any monotone network. For an arbitrary monotone Boolean network, S , $Init(S, \Pi)$ is the set of gates, u of S , such that $\Pi(u) = 1$ but for all ancestors, v of u , $\Pi(v) = 0$. Informally, we shall say that u is a *first* gate in S satisfying Π .

Similarly $Final(S, \Pi)$ is the set of gates u in S , such that $\Pi(u) = 1$ but for all descendants, v of u , $\Pi(v) = 0$. In this case, u is a *last* gate in S satisfying Π . •

Let $\mathbf{X}_{I,K}$ and $\mathbf{X}_{K,J}$ be disjoint sets of IK and KJ Boolean variables encoding the entries of two Boolean matrices $[x_{ik}]$ and $[y_{kj}]$ respectively. $BMP(\mathbf{X}_{I,K}, \mathbf{X}_{K,J})$ is the IKJ -output monotone Boolean function, having outputs $\{z_{ij} \mid 1 \leq i \leq I, 1 \leq j \leq J\}$ given by:

$$z_{ij} = \bigvee_{k=1}^K x_{ik} y_{kj}$$

Lemma 3.6: (Paterson, 1975) $\forall i \neq p, j \neq q,$

$$\text{U1) } x_{i1} \vee x_{p1} \stackrel{\text{BMP}}{\equiv} 1$$

$$\text{U2) } y_{1j} \vee y_{1q} \stackrel{\text{BMP}}{\equiv} 1$$

$$\text{U3) } x_{i1} \vee y_{1j} \stackrel{\text{BMP}}{\equiv} 1$$

Proof: We apply Lemma(3.5)(ii) for each case. It should be obvious that for all monotone functions, g ,

$$g \stackrel{\text{BMP}}{\equiv} 1 \iff \forall 1 \leq i \leq I, 1 \leq j \leq J \quad g \stackrel{z_{ij}}{\equiv} 1$$

U1) Suppose that for some monotone function, h and some r, s we have $g = (x_{i1} \vee x_{p1}) \wedge h \leq z_{rs}$. Since $i \neq p$ it must be the case that $i \neq r$ or $p \neq r$. Without loss of generality assume the former. The function z_{rs} does not depend on x_{i1} , thus $(z_{rs})^{|x_{i1} := 1} \equiv z_{rs}$. However $g^{|x_{i1} := 1} = h$ and so $h \leq z_{rs}$ as claimed.

U2) Similar to (U1)

U3) Suppose that $g = (x_{i1} \vee y_{1j}) \wedge h \leq z_{rs}$. If $i \neq r$ or $j \neq s$ then the argument used in (U1) suffices. So it may be assumed that $i = r$ and $j = s$, i.e

$$g = (x_{r1} \vee y_{1s}) \wedge h \leq z_{rs} = \bigvee_{k=1}^K x_{rk} y_{ks}$$

Thus there is some k ($1 \leq k \leq K$) for which $g \leq x_{rk} y_{ks}$. If $k \neq 1$ then $x_{rk} y_{ks}$ is independent of x_{r1} and so as in (U1) we have $h \leq x_{rk} y_{ks} \leq z_{rs}$. If $k = 1$ then,

$$g = (x_{r1} \vee y_{1s}) h \leq x_{r1} y_{1s}$$

Hence,

$$g^{x_{r1} := 1} = h \leq y_{1s} \quad ; \quad g^{y_{1s} := 1} = h \leq x_{r1}$$

and so $h = h \wedge h \leq x_{r1} \wedge y_{1s} \leq z_{rs}$. \square

Corollary 3.2:

$$\text{U4) } x_{i1} y_{1j} \vee x_{p1} y_{1j} \stackrel{\text{BMP}}{=} y_{1j}$$

$$\text{U5) } x_{i1} y_{1j} \vee x_{i1} y_{1q} \stackrel{\text{BMP}}{=} x_{i1} \quad \square$$

Theorem 3.7: (Paterson, 1975; Mehlhorn and Galil, 1976)

Any monotone Boolean network realising $BMP(\mathbf{X}_{I,K}, \mathbf{X}_{K,J})$ contains at least IJK \wedge -gates and $IJ(K-1)$ \vee -gates.

Proof: The theorem is trivial for $K = 1$. Inductively assume it holds for all values $< K$ and let S be an optimal monotone network computing $BMP(\mathbf{X}_{I,K}, \mathbf{X}_{K,J})$. Since S is minimal none of the rules U1-U5 may be applied directly to S to eliminate any gates.

With each pair ij , ($1 \leq i \leq I, 1 \leq j \leq J$) define the predicate $\Pi_{ij} : S \rightarrow \{0, 1\}$, over the nodes of S by:

$$\Pi_{ij}(u) \iff x_{i1} y_{1j} \leq \text{res}(u) \text{ and } x_{i1} \not\leq \text{res}(u) \text{ and } y_{1j} \not\leq \text{res}(u)$$

$\text{Init}(\Pi_{ij}, S)$ is the set of gates given by Defn(3.5).

Now if the sets $\text{Init}(\Pi_{ij}, S)$ are disjoint and contain only \wedge -gates, then the assignment $x_{i1} := 1 \quad \forall 1 \leq i \leq I$ eliminates at least IJ \wedge -gates from S . This is because for any $u \in \text{Init}(\Pi_{ij}, S)$ one of its inputs, v say, must have $x_{i1} \leq \text{res}(v)$.

Suppose $u \in \text{Init}(\Pi_{ij}, S)$ and has inputs from nodes v and w of S . If $\text{op}(u) = \vee$ then it must be the case that $x_{i1} \not\leq \text{res}(v)$ and $y_{1j} \not\leq \text{res}(w)$, since u satisfies Π_{ij} . It follows that $x_{i1} y_{1j} \not\leq \text{res}(v)$, and $x_{i1} y_{1j} \not\leq \text{res}(w)$ as u is a first gate which satisfies Π_{ij} . But with this

$x_{i1}y_{1j} \not\leq res(v) \vee res(w) = res(u)$, which contradicts $\Pi_{ij}(u) = 1$. Thus $op(u) = \wedge$.

Now suppose that $u \in Init(\Pi_{ij}, S) \cap Init(\Pi_{pq}, S)$, for some $p \neq i$ or $q \neq j$. If v and w are the inputs of u then either

$$x_{i1} \vee x_{p1} \leq res(v) \text{ and } y_{1j} \vee y_{1q} \leq res(w)$$

or

$$x_{i1} \vee y_{1q} \leq res(v) \text{ and } x_{p1} \vee y_{1j} \leq res(w)$$

In both cases at least one of U1-U3 can be applied and this contradicts the assumption that S is optimal. Thus the sets $Init(\Pi_{ij}, S)$ are disjoint.

We now consider the number of \vee -gates in S . For each i, j let Σ_{ij} be the predicate defined over the nodes of S by,

$$\Sigma_{ij}(u) \iff x_{i1} y_{1j} \leq res(u) \leq y_{1j} \vee \bigvee_{k \neq 1} x_{ik} \text{ and } res(u) \not\leq y_{1j}$$

It will be shown that the sets $Init(\Sigma_{ij}, S)$ are disjoint and contain only \vee -gates. This will permit IJ \vee -gates to be eliminated using the assignment $y_{1j} = 0$ for all $1 \leq j \leq J$. This is because for any $u \in Init(\Sigma_{ij}, S)$, some input, v say, of u must satisfy $res(v) \leq y_{1j}$.

Let $u \in Init(\Sigma_{ij}, S)$ and suppose $op(u) = \wedge$. If v and w are the inputs of u then

$$\begin{aligned} res(v) &\not\leq y_{1j} \vee \bigvee_{k \neq 1} x_{ik} \\ res(w) &\not\leq y_{1j} \vee \bigvee_{k \neq 1} x_{ik} \end{aligned}$$

since $u \in Init(\Sigma_{ij}, S)$. But then

$$res(u) \not\leq y_{1j} \vee \bigvee_{k \neq 1} x_{ik}$$

which contradicts $\Sigma_{ij}(u)$. It follows that u must be an \vee -gate and

$$\begin{aligned} \text{res}(v) &\leq y_{1j} \vee \bigvee_{k \neq 1} x_{ik} \\ \text{res}(w) &\leq y_{1j} \vee \bigvee_{k \neq 1} x_{ik} \end{aligned}$$

Suppose

$$u \in \text{Init}(\Sigma_{ij}, S) \cap \text{Init}(\Sigma_{pq}, S)$$

for some $(i, j) \neq (p, q)$. Using v and w as before, it must be the case that

$$\text{res}(v) \leq y_{1j} \wedge (y_{1q} \vee \bigvee_{k \neq 1} x_{pk})$$

thus $j = q$ because $x_{i1} y_{1j} \leq \text{res}(v)$. As $u \in \text{Init}(\Sigma_{ij}, S)$ we have $\text{res}(w) \not\leq y_{1j}$ and so $x_{p1} y_{1q} \not\leq \text{res}(w)$. Hence $x_{p1} y_{1q} \leq \text{res}(v)$ and thus

$$x_{i1} y_{1j} \vee x_{1p} y_{1j} \leq \text{res}(v) \leq y_{1j}$$

Now since $j = q$ it follows that $i \neq p$ and therefore from U5 $\text{res}(v) \stackrel{\text{BMP}}{\implies} y_{1j}$. So in summary this yields,

$$\begin{aligned} y_{1j} \leq \text{res}(v) \vee \text{res}(w) &\leq (y_{1j} \vee \bigvee_{k \neq 1} x_{ik}) \wedge (y_{1j} \vee \bigvee_{k \neq 1} x_{pk}) \\ &\leq y_{1j} \vee (\bigvee_{k \neq 1} x_{ik}) \wedge (\bigvee_{k \neq 1} x_{pk}) \end{aligned}$$

and now using Lemma(3.4)(i) $u \stackrel{\text{BMP}}{\implies} y_{1j}$ contradicting the optimality of S . This establishes that the sets $\text{Init}(\Sigma_{ij}, S)$ are disjoint.

The theorem now follows easily. No input can satisfy Π_{ij} or Σ_{ij} for any $1 \leq i \leq I$, $1 \leq j \leq J$, while each output z_{ij} satisfies Π_{ij} and (if $K > 1$) Σ_{ij} also. So the sets $\text{Init}(\Pi_{ij}, S)$ and $\text{Init}(\Sigma_{ij}, S)$ are non-empty and disjoint. The assignment $\langle x_{i1} = 1, y_{1j} = 0 \rangle$ eliminates the $\geq IJ$ \wedge -gates in $\text{Init}(\Pi_{ij}, S)$ and the $\geq IJ$ \vee -gates in $\text{Init}(\Sigma_{ij}, S)$. The

resulting network computes $I(K-1)J - BMP$ and so the stated lower bound on the number of \wedge -gates and \vee -gates follows from the inductive hypothesis. \square

Corollary 3.3: Any monotone network realising $IKJ - BMP$ with the minimal number of \wedge -gates and \vee -gates, computes each product $x_{ik} y_{kj}$ directly from the input nodes, and z_{ij} directly from the K products $\{ x_{ik} y_{kj} : 1 \leq k \leq K \}$.

Proof: Exercise. (See (Paterson, 1975) for solution). \square

Corollary 3.4: Let $n = N^2$. Any monotone network which computes the $\{\wedge, \vee\}$ -matrix product of $2 N \times N$ Boolean matrices contains at least $n^{3/2}$ \wedge -gates and at least $n^{3/2} - n$ \vee -gates. \square

A special case of BMP is the n -point Boolean Convolution. For this it is convenient to regard \mathbf{X}_n as the n -tuple $\langle x_0, x_1, \dots, x_{n-1} \rangle$ and \mathbf{Y}_n as the n -tuple $\langle y_0, y_1, \dots, y_{n-1} \rangle$. $CONV(\mathbf{X}_n, \mathbf{Y}_n): \{0, 1\}^{2n} \rightarrow \{0, 1\}^n$ is the function with outputs $\langle C_0, C_1, \dots, C_{n-1} \rangle$ defined by,

$$C_k(\mathbf{X}_n, \mathbf{Y}_n) = \bigvee_{i+j \equiv k \pmod{n}} x_i y_j$$

This is the *cyclic* convolution; a variant is the *shifting* convolution, $SH(\mathbf{X}_n, \mathbf{Y}_n)$, which has $2n - 1$ outputs, $\langle SH_0, \dots, SH_{2n-2} \rangle$ defined by,

$$SH_k = \bigvee_{i+j=k} x_i y_j$$

The shifting convolution is a special case of integer multiplication. All known lower bounds for $CONV$ can be shown to hold for SH using essentially identical arguments.

$CONV$ has been examined by a number of authors. (Pippenger and Valiant, 1976) and independently Lamagna (1979) obtained lower

bounds of $\Omega(n \log n)$ on its monotone complexity. Blum (1984b) gives a lower bound of $\Omega(n^{4/3})$ on the number of \wedge -gates required. Blum's methods are extremely complicated and cannot be presented here. The best bound attained, to date, is that of Weiss (1984): $\Omega(n^{3/2})$ on the number of \vee -gates. This is described below. The structure of optimal monotone networks realising *BMP* motivates the following

Conjecture 3.1: Any monotone network realising *CONV* contains at least n^2 \wedge -gates and at least $n^2 - n$ \vee -gates. •

This, and its weaker form $\mathbf{C}^m(\text{CONV}) = ? \Omega(n^2)$, remains unresolved.

Weiss' result is based on an elegant "information-flow" argument which identifies a set of \sqrt{n} distinct \vee -gates all of which may be eliminated by fixing $x_{n-1} := 0$. The inductive argument is applied to those functions, $QUAD(\mathbf{X}_n, \langle y_0 \cdots y_k \rangle) \in M_{n+k,m}$ which satisfy:

A1) For every output function Q_i of *QUAD* and each input variable, z , at most one prime implicant of Q_i depends on z .

A2) $\forall 1 \leq i, j \leq m \quad |Q_i \cap Q_j| \geq 1 \iff i = j$

A3) $\forall 1 \leq i \leq m \quad p \in \mathbf{PI}(Q_i) \iff p = x_r y_s$ for some $1 \leq r \leq n, 1 \leq s \leq k$.

Clearly the sequence of n functions

$$[\text{CONV}_{n-i}(\mathbf{X}_n - \{x_{n-1}, \dots, x_{n-i}\}, \mathbf{Y}_n)]_{i=0}^{i=n-1}$$

defined by:

$$\text{CONV}_i = \begin{cases} \text{CONV}(\mathbf{X}_n, \mathbf{Y}_n) & \text{if } i = n \\ \text{CONV}_{i+1}(\mathbf{X}_n - \bigcup_{j=i+1}^{n-1} x_j, \mathbf{Y}_n) |_{x_i := 0} & \end{cases} \quad (3.11)$$

satisfies (A1-A3). The lower bound of Weiss (1984) is established by

showing that \sqrt{n} \vee -gates can be eliminated by the assignment which renders a monotone network realising $CONV_i$ into one computing $CONV_{i-1}$.

Below $C_{\vee}^m(f)$ denotes the minimal number of \vee -gates needed in any monotone network realising $f \in M_n$.

Theorem 3.8: (Weiss, 1984) $\forall n \geq 1$

$$C_{\vee}^m(CONV(\mathbf{X}_n, \mathbf{Y}_n)) = \Omega(n^{3/2})$$

Proof: We use induction over i to prove,

$$\forall 1 \leq i \leq n \quad C_{\vee}^m(CONV_i(\mathbf{X}_n - \bigcup_{j=i}^{n-1} x_j, \mathbf{Y}_n)) \geq i \sqrt{n}$$

and then from (3.11) this establishes the theorem.

The inductive base $i = 1$ is immediate. Inductively assume that the lower bound on $C_{\vee}^m(CONV_j)$ holds for all values $1 \leq j \leq i$. Let S be an optimal monotone network realising $CONV_{i+1}$ at nodes $\langle t_0, t_1, \dots, t_{n-1} \rangle$, so that:

$$res(t_k) = C_k^{|\{x_j := 0 : i+1 \leq j \leq n-1\}|}$$

Note that since $i+1 \geq 2$ each output function has at least 2 prime implicants. Consider the input x_i of S . From the definition of $CONV_{i+1}$, it follows that every output function $res(t_k)$ for $0 \leq k \leq n-1$ has exactly one prime implicant $x_i y_{(k-i) \bmod n}$ which depends on x_i . Let $\sigma_i G(S) \rightarrow \{0, 1\}$ be a predicate defined over the gates G of S by;

$$\begin{aligned} op(u) &= \vee, u \text{ lies on a path } [x_i, t_k] \\ \sigma(u) &\iff \text{and} \\ &\exists j \neq i, h \text{ s.t. } x_j y_h \leq res(u) \end{aligned}$$

As before $Init(\sigma, S)$ are the first \vee -gates satisfying σ in S . Thus $u \in Init(\sigma, S)$ if and only if u is the first \vee -gate on a path from x_i to an output for which $x_j y_h \leq res(u)$, with $j \neq i$. We first show that every path from x_i to an output contains a gate $u \in Init(\sigma, S)$. Suppose the contrary, so that there is some path from x_i to an output t_k which is devoid of gates satisfying σ . Let $v_1, v_2, \dots, v_r \equiv t_k$ be the gates on this path. Since no \vee -gate on this path satisfies σ one of the following conditions must hold:

C1) $op(v_q) = \wedge \forall 1 \leq q \leq r$.

C2) If $op(v_q) = \vee$ then $\forall m \in \mathbf{PI}(res(v_q))$ either $m \equiv x_i y_j$ or $\neg \exists p$ such that $m p \in \mathbf{PI}(CONV_{i+1})$.

Now (C1) cannot hold since then setting $x_i = 0$ renders $res(t_k)$ equal to 0, which contradicts the definition of $CONV_i$. Thus it may be assumed that the path under consideration contains some \vee -gates and that these must satisfy (C2). Let v_q be any \vee -gate on this path. If x_i is fixed to 0, then from (C2) and Lemma(3.4)(i) we have

$res(v_q)^{|x_i := 0} \stackrel{CONV_{i+1}}{\equiv} 0$. It follows that under the partial

assignment $x_i := 0$ every gate on the path x_i to $v_r \equiv t_k$ becomes 0, since it is either an \vee -gate replaceable by 0, or an \wedge -gate one of whose inputs is 0. Again this contradicts the definition of $CONV_i$ and thus every path from x_i to an output contains a gate, u , satisfying σ , and so a gate in $Init(\sigma, S)$.

From the previous paragraph we can conclude that the assignment $x_i := 0$ allows every gate in $Init(\sigma, S)$ to be eliminated. To prove the theorem it remains to show that $|Init(\sigma, S)| \geq \sqrt{n}$. Then, since all these gates are \vee -gates and $CONV_{i+1}^{|x_i := 0} \equiv CONV_i$ the lower bound asserted follows from the inductive hypothesis.

Suppose that $E = |Init(\sigma, S)| < \sqrt{n}$. Let u_1, \dots, u_E be the gates in $Init(\sigma, S)$. By the definition of σ for each u_q there exists some $j_q \neq i$ and h_q for which $x_{j_q} y_{h_q} \leq res(u_q)$. Consider the partial assignment $\{x_{j_q} := 1, y_{h_q} := 1 : \forall 1 \leq q \leq E\}$. Under this all the gates u_q take the value 1. The resulting network is therefore independent of x_i , since some u_q lies on each path from x_i to any output t_k . Consider the function computed by each output t_k under this assignment. All of these must be independent of x_i . Before the assignment $x_i y_{k_i}$ is a prime implicant of $res(t_k)$. $res(t_k)$ becomes independent of x_i if and only if it becomes the constant 1 or this prime implicant becomes y_{k_i} . However this would imply that $res(t_k)$ had 2 prime implicants depending on y_{k_i} (because x_i is not fixed to 1, so the only way in which this latter case could arise is for $x_r y_{k_i}$ to be a prime implicant of $res(t_k)$ with x_r being a variable fixed to 1). It follows that all the n outputs whose result depends on x_i must become 1 under this partial assignment. But now a contradiction results since no more than $E^2 < n$ outputs can have some prime implicant satisfied by the given assignment. This proves that $E \geq \sqrt{n}$ and so the theorem is established. \square

The structure of *BMP* allows replacement rules to be used directly to deduce properties of optimal monotone networks: gates realising certain simple functions cannot occur because they are replaceable by constants or some input variable. However applications in this pure form are not available for many sets of functions. For example one could obtain a lower bound of $\Omega(n^2)$ on the number of \wedge -gates required to compute *CONV* if the replacements $x_i \vee x_j \stackrel{CONV}{=} 1$ and $y_i \vee y_j \stackrel{CONV}{=} 1$ were valid for $i \neq j$ ^{f)}. As with *CONV* for many functions in $M_{n,m}$ we can identify specific functions which, if replaceable by constants or input variables,

f) These are not valid, cf Lemma(3.5)(iii)

allow good lower bounds to be proved since sufficient information about the form of optimal networks is made available by their absence.

In the next result presented this difficulty arises but is circumvented by employing a technical device: it is assumed that certain functions are available as additional inputs, the cost of computing these is assumed to be 0 as far as deriving the required lower bounds is concerned. Thus certain useful functions are provided for "free". Let $\mathbf{C}^{\mathbf{m}^*}(f)$ denote the number of 2-input \wedge and \vee -gates required to compute $f \in M_n$ by a network in which the inputs are $\mathbf{X}_n \cup \langle g_1, \dots, g_p \rangle$ for some set of monotone Boolean functions g_1, g_2 etc. Obviously $\mathbf{C}^{\mathbf{m}^*}(f) \leq \mathbf{C}^{\mathbf{m}}(f)$, so any lower bound on $\mathbf{C}^{\mathbf{m}^*}$ is trivially a lower bound on $\mathbf{C}^{\mathbf{m}}$. By providing additional inputs it becomes possible to employ a wider variety of replacement rules, for now no optimal network can contain a *gate* which computes one of the functions g_i or a function which is replaceable by g_i . The technique of providing functions for free as additional inputs is one which had previously been applied in the sphere of algebraic complexity. Wegener (1980) is one of the earliest instances of its application to Boolean networks. This paper proves lower bounds on the size of monotone networks realising certain sets of Boolean sums.

Definition 3.6: $F(\mathbf{X}_n) \equiv \langle s_1, s_2, \dots, s_m \rangle \in M_{n,m}$ is a set of Boolean sums if each function $s_i \in F$ satisfies

$$\forall p \in \mathbf{PI}(s_i) \quad p \in \mathbf{X}_n$$

For any such $\langle s_1, \dots, s_m \rangle$ we shall use P_i to denote the subset of \mathbf{X}_n on which s_i essentially depends.

For $1 \leq h \leq m-1$, $0 \leq k \leq n$ a set of m Boolean sums F is (h,k) -disjoint if and only if,

$\forall \{i_0, i_2, \dots, i_h\} \subset \{1, 2, \dots, m\}$

$$|\bigcap_{j=0}^h P_{i_j}| \leq k \quad \bullet$$

Neciporuk (1970) proved that any set of m (1,1)-disjoint sums, F , has monotone network complexity,

$$\mathbf{C}^m(F) = \sum_{i=1}^m (|P_i| - 1)$$

i.e the obvious network using only \vee -gates is optimal. Neciporuk defined specific instances in $M_{n,n}$ having complexity $\Omega(n^{3/2})$. Wegener (1979) generalised this result to arbitrary (1, k)-disjoint sums. The result we now give is from Mehlhorn (1979) and applies to any set of (h, k)-disjoint sums. Mehlhorn's approach employs the method of providing additional functions, at no cost, as extra inputs. The key ideas behind the proof lie in the fact that it would be easy to determine a lower bound on the complexity of (h, k)-disjoint sums *if* it could be assumed that optimal networks contained only \vee -gates. In general this assumption is invalid, since there are sets of Boolean sums for which minimal size monotone networks contain \wedge -gates. Wegener (1979) proved that optimal monotone networks computing (1, k)-disjoint sums having all Boolean sums of $\leq k$ variables provided free, do not contain any \wedge -gates. Mehlhorn (1979) develops this result by proving that \wedge -gates cannot help reduce network size significantly for similar networks computing (h, k)-disjoint sums. We employ the following notation.

A $k - \Sigma$ network is a monotone network in which all sums containing $\leq k$ variables are provided as additional inputs. Let $\Omega \subseteq \{\wedge, \vee\}$ and F be a set of m (h, k)-disjoint sums,

$C_{\Omega}^{\mathbf{m}^*}(F)$ = Size of a minimal $k - \Sigma$ network realising F over the basis Ω

$C_{\Omega}^{\mathbf{m}}(F)$ = Size of minimal network realising F over the basis Ω

Lemma 3.7: (Mehlhorn, 1979) Let $F(\mathbf{X}_n) \{0, 1\}^n \rightarrow \{0, 1\}^m$ be a set of m (h, k) -disjoint sums.

$$\text{i) } C_{\vee}^{\mathbf{m}^*}(F) \leq \max \{1, h - 1\} C_{\wedge, \vee}^{\mathbf{m}^*}(f)$$

$$\text{ii) } C_{\vee}^{\mathbf{m}}(F) \leq \max \{1, h - 1, k - 1\} C^{\mathbf{m}}(f)$$

Proof: i) Let S be an optimal $k - \Sigma$ network over the basis $\{\wedge, \vee\}$ realising F . S contains d \vee -gates and c \wedge -gates so that $C^{\mathbf{m}^*}(F) = c + d$. It is proved that for each $0 \leq i \leq c$ there is a $k - \Sigma$ network, S_i , which contains at most $c - i$ \wedge -gates and at most $d + (h - 1)i$ \vee -gates. The inductive base $i = 0$ easily follows by choosing $S_0 \equiv S$. Now assume the assertion hold for all values $\leq i$ and let S_i be a $k - \Sigma$ network realising F and containing $\leq c - i$ \wedge -gates and $\leq d + (h - 1)i$ \vee -gates. It may be assumed that S_i contains at least one \wedge -gate otherwise the inductive step is immediate. Let u be a last \wedge -gate in S_i , and v, w the nodes which supply the inputs of u . In this case;

$$\begin{aligned} \text{res}(u)(\mathbf{X}_n) &= \text{res}(v)(\mathbf{X}_n) \wedge \text{res}(w)(\mathbf{X}_n) \\ &= y_1 \vee y_2 \vee \cdots \vee y_p \vee m_1 \vee \cdots \vee m_q \end{aligned}$$

where $y_j \in \mathbf{X}_n$ for each $1 \leq j \leq p$ and each m_j is a product of at least 2 variables from \mathbf{X}_n .

There are two cases,

Case 1: $p \leq k$. The sum $\bigvee_{j=1}^p y_p$ is available as an input function of S_i .

The function $\bigvee_{j=1}^q m_j$ is replaceable by 0. It follows that u may be

eliminated and replaced by an input of S_i . The resulting network is S_{i+1} , and it contains no additional \vee -gates and one fewer \wedge -gate.

Case 2: $p > k$. Without loss of generality let $\{t_1, \dots, t_r\}$ be the output gates such that there is a path from u to t_j , for each $1 \leq j \leq r$. Then

$$\begin{aligned} s_j &= \text{res}(t_j) = \text{res}(u) \vee b_j \\ &= y_1 \vee \dots \vee y_p \vee \text{res}(v)\text{res}(w) \vee b_j \end{aligned}$$

for some function $b_j \in M_n$, $b_j \not\equiv 1$. From this it follows that $r \leq h$ for $p > k$ and F is (h, k) -disjoint. We claim that $\forall 1 \leq j \leq r$

$$s_j = \text{res}(v) \vee b_j \quad \text{or} \quad s_j = \text{res}(w) \vee b_j$$

Obviously

$$s_j = \text{res}(u) \vee b_j = (\text{res}(v) \vee b_j)(\text{res}(w) \vee b_j)$$

So to prove the claim it is sufficient to show that $\text{res}(v) \leq s_j$ or $\text{res}(w) \leq s_j$. Suppose for some j neither of these is true. In this case there are prime implicants, q of $\text{res}(v)$ and r of $\text{res}(w)$ such that,

$$|\text{var}(q) \cap P_j| = |\text{var}(r) \cap P_j| = 0$$

But then $qr \not\leq s_j$, although $qr \leq \text{res}(u)$, so $\text{res}(u) \not\leq s_j$, and this contradiction proves the claim.

We can now construct S_{i+1} . Replace u by the constant 0 so that $\text{res}(t_j) := b_j$ for each $1 \leq j \leq r \leq h$; this must eliminate at least one \vee -gate from S_i unless $r = 1$ and $\text{res}(u) = s_1$, in which case $s_1 = \text{res}(v)$ or $s_1 = \text{res}(w)$ and the proof is complete. Otherwise, add r \vee -gates g_1, \dots, g_r to S_i ; the gate g_j having inputs t_j and v if $s_j = b_j \vee \text{res}(v)$; t_j and w if $s_j = b_j \vee \text{res}(w)$. S_{i+1} still realises F , contains one fewer \wedge -gate at at most $h - 1$ additional \vee -gates. This

completes the inductive step. The network S_c contains only \vee -gates and has size $\leq \max\{1, h-1\} \mathbf{C}^{\mathbf{m}^*}(F)$.

ii) The method of (i) is used to convert a monotone network, S , containing c \wedge -gates and d \vee -gates to a sequence $S = S_0, S_1, \dots, S_c, S_i$ containing at most $c-i$ \wedge -gates and no more than $d + \max\{h-1, k-1\} \cdot i$ \vee -gates. The only difference occurs in the $p \leq k$ case where the sum $y_1 \vee \dots \vee y_p$, not being free, must be computed directly using $p-1$ \vee -gates. \square

Theorem 3.9: For any set of (h, k) -disjoint sums, F ,

$$\mathbf{C}_{\vee}^{\mathbf{m}}(F) \geq \mathbf{C}_{\vee}^{\mathbf{m}^*}(F) \geq \frac{\sum_{i=1}^m (\lceil \frac{|P_i|}{k} \rceil - 1)}{h}$$

Proof: The first inequality is obvious. For the second consider an optimal k - Σ network over the basis $\{\vee\}$ realising F , with output gates $\langle t_1, \dots, t_m \rangle$. Since at most k variables occur in any sum provided as a free input, there is a path from at least $\lceil \frac{|P_i|}{k} \rceil$ inputs to the output gate t_i , for each $1 \leq i \leq m$. If u is any \vee -gate in S , then from the optimality assumption, it follows that $\text{res}(u)$ is a sum of at least $k+1$ variables and hence there is a path from u to no more than h output gates. For each gate u of S , let $\delta(u)$ denote the number of outputs t_i such that there is a path from u to t_i . From the preceding argument clearly,

$$\sum_{u \in G(S)} \delta(u) \leq h \cdot \mathbf{C}_{\vee}^{\mathbf{m}^*}(F)$$

On the other hand the number of gates, v , such that there is a path from v to t_i must be at least $\lceil \frac{|P_i|}{k} \rceil - 1$ since there is a path from at least $\lceil \frac{|P_i|}{k} \rceil$ inputs to t_i . This gives,

$$\sum_{u \in G(S)} \delta(u) \geq \sum_{i=1}^m (\lceil \frac{|P_i|}{k} \rceil - 1)$$

proving the theorem. \square

Corollary 3.5 For any set F of m (h, k) -disjoint Boolean sums,

$$C^m(F) \geq \frac{\max \{1, h-1\} \sum_{i=1}^m (\frac{|P_i|}{k} - 1)}{h}$$

\square

Mehlhorn (1979) defines an explicit set of Boolean sums having monotone complexity $\Omega(n^{5/3})$. The set defined uses a result of Brown (1966) and the fact the Boolean sums may be represented as bipartite graphs over disjoint sets $\{s_1, \dots, s_m\}$ and $\{x_1, \dots, x_n\}$ of vertices. In such graphs there is an edge between x_i and s_j if and only if $x_i \in P_j$. A set of sums is (h, k) -disjoint if and only if the bipartite graph defined thus does not contains $K_{h-1, k-1}$ as a subgraph, i.e the complete bipartite graph on two sets of h and k vertices. Brown (1966) constructs an $2n$ -vertex bipartite graph having $\Omega(n^{5/3})$ edges and without $K_{3,3}$ as a subgraph. This graph defines a set of $(2,2)$ -disjoint Boolean sums having complexity $\Omega(n^{5/3})$.

To conclude this section we outline a method first applied successfully in Wegener (1982) to obtain lower bounds of size $\Omega\left(\frac{n^2}{\log n}\right)$.

The paper introduces the concept of *value functions*.

Let $F = \langle f_1, f_2, \dots, f_m \rangle (\mathbf{X}_n) \in M_{n,m}$. Given an optimal monotone network, S , realising F suppose that for each \wedge -gate, u of S , a function $V_u: \mathbf{PI}(F) \rightarrow [0, 1]$ is defined which satisfies;

$$\sum_{p \in \mathbf{PI}(F)} V_u(p) \leq 1. \text{ Then;}$$

$$\sum_{\{u : op(u) = \wedge \text{ in } S\}} \sum_{p \in \mathbf{PI}(F)} V_u(p) \leq \mathbf{C}^m(F)$$

Thus,

$$\mathbf{C}^m(F) \geq \sum_{p \in \mathbf{PI}(F)} \sum_{\{u : op(u) = \wedge \text{ in } S\}} V_u(p)$$

So if for every prime implicant, p , of F it holds that

$$\sum_{\{u : op(u) = \wedge \text{ in } S\}} V_u(p) \geq h(n+m)$$

for some function $h: \mathbf{N} \rightarrow \mathbf{R}^+$ then,

$$\mathbf{C}^m(F) \geq \frac{|\mathbf{PI}(F)|}{h(n+m)}$$

Wegener (1982) defines the set of functions f_{MN}^m , which have mMN inputs and M^m outputs. The inputs correspond to the entries of m $M \times N$ Boolean matrices. The outputs, $\{y_{h_1 \dots h_m} : 1 \leq h_1, \dots, h_m \leq M\}$ are given by,

$$y_{h_1 \dots h_m} = \bigvee_{l=1}^N x_{h_1 l}^1 x_{h_2 l}^2 \cdots x_{h_m l}^m$$

x_{ij}^k denoting the ij -th entry of the k -th matrix. This set of functions is known as the *Direct Matrix Product* ($mMN - DMP$). An output is 1 if and only if all the matrix rows referenced have a common 1. Wegener (1982) combines the techniques of providing certain functions for free (specifically all products of fewer than m variables) and the use of value functions to prove;

$$\mathbf{C}^m(mMN - DMP) \geq \frac{NM^m}{2}$$

The paper defines specific values of m , M and N for which

$$C^m(mMN - DMP) = \Omega\left(\frac{n^2}{\log n}\right)$$

where $n \geq \max\{mMN, M^m\}$.

3.4) Linear Lower Bounds on the Monotone Complexity of Single-Output Functions

The methods used to obtain lower bounds for functions in $M_{n,m}$ have not proved adaptable to networks realising single output monotone Boolean functions. One reason for this is the fact that the techniques used rely to some extent on "information-flow" arguments; the idea that certain properties may be deduced from the knowledge that several outputs depend on a single gate of the lower bound for Boolean Convolution and that for sets of (h, k) -disjoint sums. In the next section some techniques for deriving superpolynomial bounds on functions in M_n are presented. In this section we describe some earlier results exhibiting linear lower bounds. First a lower bound of $2.5n - 5.5$ on the monotone complexity of $T_k^n(\mathbf{X}_n)$, for $3 \leq k \leq n - 2$ is proved. We then outline a lower bound of $3.5n$ on the monotone complexity of the majority function and present the $4n$ lower bound of Tiekhenrich (1984). To conclude an upper bound of kn for T_k^n is established, when k is fixed.

In Dunne (1985a) the following result is proved.

$$C^m(T_k^n) \geq 2.5n - 5.5 \quad \text{for } n \geq k \text{ and } 3 \leq k \leq n - 2$$

It is sufficient to consider only the case $k = 3$, since for $4 \leq k \leq \lfloor n/2 \rfloor$ it will be clear that the same proof is applicable, and for $\lceil n/2 \rceil \leq k \leq n - 2$, the relation $\tilde{T}_k^n = T_{n-k+1}^n$ establishes the result by duality.

In common with the lower bounds on combinational complexity presented earlier, the method used combines an inductive analysis of optimal monotone networks with a counting argument. For the inductive stage only partial assignments which set inputs to 0 are usable. To prove similar or larger bounds by setting an input to 1 would require at least $\lfloor n/2 \rfloor$ gates to be eliminated. Some preliminary results are required for the lower bound proof.

Lemma 3.8: Let S be an optimal monotone network computing T_k^n at some node t . S does not contain any gate g for which:

$$T_{k_1}^n \leq \text{res}(g) \quad \forall 1 \leq k_1 < k \text{ and } k \geq 2$$

Proof: Suppose S contains a gate g such that $T_{k_1}^n \leq \text{res}(g)$ for some k_1 as above. We shall show that S is not optimal. Let \tilde{S} denote the monotone dual network of S . This network computes T_{n-k+1}^n . Let \tilde{g} be the dual function of $\text{res}(g)$ computed in \tilde{S} . Clearly $\tilde{g} \leq T_{n-k_1+1}^n$. By Lemma(3.4)(i) g in \tilde{S} is replaceable by the constant 0. Thus, by duality, g in S is replaceable by the constant 1. It follows that S was not optimal. \square

Lemma 3.9: There is an optimal monotone network S computing T_k^n , such that every input x_i of S which has fan-out equal to 1, enters an \wedge -gate.

Proof: We show how to restructure S to a network S^* satisfying the lemma. Let x_i be an input of S having fan-out equal to 1 and entering an \vee -gate g whose other input is some function f . Observe that $f \leq T_k^n$. For suppose $\{x_{p_1}, \dots, x_{p_{k-1}}\}$ is a subset of \mathbf{X}_n such that the monom formed by \wedge -ing the variables in this set is an implicant of f . The partial assignment $x_{p_j} := 1, \forall 1 \leq j \leq k-1$ leaves S independent of x_i , but under this assignment S should compute

$$T_1^{n-k+1}(\mathbf{X}_n - \bigcup_{j=1}^{k-1} \{x_{p_j}\})$$

which depends on x_i . This contradiction establishes that every prime implicant of f is an implicant of T_k^n . Now, since $g \neq t$, the output gate, S can be restructured as follows:

- 1) Replace gate g in S by the input x_i .
- 2) Add one \vee -gate to S with inputs f and the output of t

Clearly the new network contains no more gates than S , and computes T_k^n . If g has only a single \vee -gate as successor then the steps above may be repeated. Eventually the fan-out of x_i must increase or x_i must enter an \wedge -gate. As the fan-out of other inputs is not affected, this process may be applied repeatedly until the lemma is true for all inputs. \square

Lemma 3.10: Let S be any monotone network which computes T_k^n (where $n > k$). Let x_i be any input of S which enters exactly 2 \vee -gates, whose other inputs are f_1, f_2 . For each r with $2 \leq r \leq k-1$ if there exists any monom m_1 over $\mathbf{X}_n - \{x_i\}$ such that

$$m_1 \leq f_1 \wedge T_{k-r}^{n-1}(\mathbf{X}_n - \{x_i\})$$

then there does not exist any monom m_2 over $\mathbf{X}_n - \{x_i\}$ such that

$$m_2 \leq f_2 \wedge T_{r-1}^{n-1}(\mathbf{X}_n - \{x_i\})$$

Proof: Suppose m_1 and m_2 are two such monoms. The partial assignment $x_j := 1 \forall x_j \in \text{var}(m_1) \cup \text{var}(m_2)$ leaves S independent of x_i . But under the assignment S should compute

$$T_{k-q}^{n-q}(\mathbf{X}_n - \text{var}(m_1) - \text{var}(m_2))$$

($q = |\text{var}(m_1) \cup \text{var}(m_2)|$) and this depends on x_i since $q \leq k-1$. This contradiction proves the lemma. \square

Theorem 3.10: (Dunne, 1985a)

$$C^m (T_3^n (\mathbf{X}_n)) \geq 2.5n - 5.5$$

Proof: By induction on $n \geq 3$. The inductive base is obvious, so assume the theorem holds for all values $< n$ and let S be an optimal monotone network realising T_3^n at a unique node t . We proceed by a case analysis. It is assumed that S has been subjected to the process of Lemma(3.9) and thus any input having fan-out equal to 1 enters an \wedge -gate in S . The cases

Case 1: $\exists x_i \in \mathbf{X}_n$ such that $\phi(x_i) \geq 3$.

Case 2: $\exists x_i \in \mathbf{X}_n$ such that $\phi(x_i) = 2$ and x_i enters an \wedge -gate. (Figure(3.2)).

Figure 3.2

are straightforward and it is left to the reader to confirm that 3 gates may be eliminated from S by fixing x_i to 0.

Case 3: $\exists x_i \in \mathbf{X}_n$ such that $\phi(x_i) = 1$.

x_i enters some \wedge -gate, g say. Let h be the gate which supplies the other input of g . It is easy to see that $g \neq t$ and $T_2^{n-1}(\mathbf{X}_n - \{x_i\}) \leq \text{res}(h)$. Setting $x_i=0$ eliminates g and its successor. The resulting network computes $T_3^{n-1}(\mathbf{X}_n - \{x_i\})$, but still contains gate h , with $T_2^{n-1}(\mathbf{X}_n - \{x_i\}) \leq \text{res}(h)$. From Lemma(3.8) the gate h may be replaced by 1 in this network. Thus setting $x_i = 0$ eliminates 3 gates.

This leaves only,

Case 4: $\forall x_i \in \mathbf{X}_n$ x_i enters exactly 2 \vee -gates. (Figure(3.3))

Figure 3.3

Case 4.1: $\exists x_i, x_j$ such that x_i, x_j enter an \vee -gate g and $\phi(g) > 1$ or g enters an \wedge -gate. At least 5 gates may be eliminated by setting $x_i = x_j = 0$. This would be sufficient to prove the result.

To summarise it may now be assumed that:

- A1) Every network input enters exactly 2 \vee -gates, g_1, g_2
- A2) From Lemma(3.10): for at most one of the functions f_1, f_2 which enter these gates is it true that there exists x_k such that $x_k \leq f_i$ ($i = 1$ or 2)
- A3) If g_1 has inputs x_i and x_j then g_1 has only one immediate successor and this is an \vee -gate.

For any T_3^n network which is not of this form sufficient gates can be eliminated to apply the inductive argument.

The lower bound for the remaining case is derived by a wire counting argument, without recourse to the inductive hypothesis. Let:

$out(Q) = | \{ \text{The set of wires out of a set of nodes } Q \} |$

$T = \{ \vee\text{-gates } g : x_i \text{ is an input of } g \text{ and } \exists j \neq i \text{ s.t. } x_j \leq res(g) \}$

$R = \{ \vee\text{-gates } g : x_i \text{ is an input of } g, g \notin T \}$

$T_1 = \{ \vee\text{-gates } g \in T : x_i, x_j \text{ are inputs of } g \}$

$T_2 = T - T_1$

$M = \{ \vee\text{-gates } g : g \text{ is the unique successor of some } h \in T_1, g \notin T_2 \}$

$U = \{ \vee\text{-gates } g \in T_2 : g \text{ is the unique successor of some } h \in T_1 \}$

$E = \{ g : g \notin T \cup R \cup M \}$

We can observe the following:

B1) $out(\mathbf{X}_n) = 2n$ (By analysis above).

B2) $out(R) \geq |R|$ (By optimality of S).

B3) $out(T) \geq |T|$ (By optimality of S).

B4) $out(T_1) = |T_1| = |U| + |M|$. This holds as each gate in U has only one input from a gate in T_1 . Although a gate in M may have two inputs from gates in T_1 , since T_1 gates have fanout=1, by (A3), S may be restructured in this case so that each gate in M has only one input from a T_1 gate. (Fig(3.4))

Figure 3.4

B5) $out(E) \geq |R|$ (By (A2), as each gate in R must have one input from a gate not in $R \cup T \cup M$).

B6) $2|T_1| + |T_2| + |R| = out(\mathbf{X}_n)$

B7) $|T_1| + |T_2| = |T|$ (By definition).

B8) $out(M) \geq |M|$ (By optimality).

Now, it is clear that for any network S

$$C^m(S) = 1/2 out(\mathbf{X}_n \cup S)$$

To prove $2.5n$ the lower bound given by (B2) must be improved by showing that:

$$out(R) \geq |R| + |U|$$

Definition 3.7: Let S compute T_3^n . A U -configuration is a subnetwork α of S consisting of 5 gates $\{g_i, g_j, g_k, g_4, g_5\}$ arranged as in Figure(3.5). •

Lemma 3.11: Let $P = \{i, j, k\}$ and let S be an optimal monotone network computing T_3^n . S may be restructured to a monotone network S^* which is no larger than S , computes T_3^n and satisfies:

(*) For each U -configuration in S^* , there exists some $p \in P$ such that every path from g_p to an \wedge -gate splits, i.e there exists a gate u on a path from g_p to an \wedge -gate h such that $\phi(u) > 1$.

Proof: Suppose S does not satisfy the lemma. Let α be any U -configuration for which (*) is false. Let h_i, h_j, h_k be the first \wedge -gate encountered on paths from g_i, g_j, g_k . (Note that there can only be one "first" \wedge -gate on each path as no path splits). All the gates on the paths $[g_p, h_p]$ are \vee -gates. Let F_i, F_j, F_k be the function \vee -ed with x_i, x_j, x_k on these paths. Let B_i, B_j, B_k be the function fed to the other input of h_i, h_j, h_k , so that $res(h_p) = B_p(F_p \vee x_p)$. We perform one modification.

C) If $x_p \leq B_p$ then compute $(x_p \vee F_p) B_p$ by using one \wedge -gate to compute $F_p B_p$ and \vee the result with x_p . h_p and g_p can then be eliminated. (Fig(3.6))

Figure 3.5

Thus we may assume that $\forall p \in \{i, j, k\} x_p \not\leq B_p$

We now prove three properties of this subnetwork.

Property 1: h_i, h_j and h_k are distinct.

Proof: Suppose, without loss of generality, that $h_i=h_j$, so that $B_i = x_j \vee F_j$ and $B_j = x_i \vee F_i$, as in Figure(3.7). Consider the assignment $x_k=1$. By arguments similar to the proof of Lemma(3.10) it is easy to see that

$$x_k x_l \not\leq F_i \vee F_j \quad \forall x_l \in \mathbf{X}_n - \{x_i, x_j\}$$

Thus $S^{|x_k=1}$ depends on x_i, x_j only via h_i . This implies that all gates

Figure 3.6

whose result depends on x_p , other than those on the path $[g_p, h_p)$ are descendants of h_i ($p = i$ or j). But $res(h_i) = (x_j \vee F_j)(x_i \vee F_i)$ and the only prime implicants of this function involving x_i or x_j have the form $x_i x_j$ or $x_i x_p x_q$ or $x_j x_p x_q$ where $p \neq q$. Therefore $S^{|x_k=1}$ cannot compute $T_2^{n-1}(\mathbf{X}_n - \{x_k\})$ and this contradiction establishes the property. \square

Figure 3.7

Property 2: Let g be a gate of S such that:

b1) $x_i x_j x_k \leq res(g)$

b2) $\forall p \in \{i, j, k\}$ g is not a descendant of any gate on a path $[g_p, h_p]$.

Then: $x_i \vee x_j \vee x_k \leq res(g)$

Proof: All such gates are descendants of g_5 . Partition these descendants into sets according to their distance from g_5 , e.g. By breadth-first search rooted at g_5 . The proof proceeds by induction on d , the distance of sets from g_5 . The base $d=0$ is obvious, as the only gate involved is g_5 itself. For the inductive step assume that Property(2) is true for all gates at distance less than d from g_5 and let g be a gate at distance d from g_5 such that $x_i x_j x_k \leq res(g)$. Let g' and g'' be the inputs of g , both of which satisfy (b2). If g is an \vee -gate then

$x_i x_j x_k \leq \text{res}(g')$ or $x_i x_j x_k \leq \text{res}(g'')$, without loss of generality suppose the former. Since the distance of g' from g_5 is less than d , by the inductive hypothesis, $x_i \vee x_j \vee x_k \leq \text{res}(g')$, and so by monotonicity $x_i \vee x_j \vee x_k \leq \text{res}(g)$. If g is an \wedge -gate then $x_i x_j x_k \leq \text{res}(g')$ and $x_i x_j x_k \leq \text{res}(g'')$ and this case follows by a similar argument. \square

Property 3: For all $p \in \{i, j, k\}$ $x_i x_j x_k \not\leq B_p$

Proof: Suppose, without loss of generality, that $x_i x_j x_k \leq B_i$. The gate which computes B_i must be a descendant of h_j or h_k . To see this recall that $h_i \not\leq h_j$ and $h_i \not\leq h_k$ (Property(1)), and so if this observation were false, Property(2) would apply and $x_i \vee x_j \vee x_k \leq B_i$ contradicting the modification (C). It follows that h_i is a descendant of h_j (or h_k) and thus $x_i x_j x_k \leq B_j$ (or B_k). By repeating the argument twice a cycle in S would result. This contradiction proves the claim. \square

Lemma(3.11) now follows easily for consider the partial assignment $\mathbf{X}_n - \{x_i, x_j, x_k\} = 0$. Then from Property(3) $B_p = 0, \forall p \in \{i, j, k\}$. S under this partial assignment cannot compute $T_3^3(x_i, x_j, x_k)$ as it only depends on x_i, x_j and x_k via g_5 which computes $T_1^3(x_i, x_j, x_k)$. Contradiction. \square

From this Lemma it follows that $\text{out}(R) \geq |R| + |U|$. For let α be any U -configuration in S . Without loss of generality suppose a path from g_i in α splits before meeting an \wedge -gate. Let F_i be the function \vee -ed with x_i on this path before it splits. It is clear that S may be restructured in such a way that x_i enters an \vee -gate g whose other input is F_i with $\phi(g) \geq 2$. This may be done without increasing the size of S , and for all U -configurations in S . (Figure(3.8))

This now gives:

$$\text{out}(S \cup \mathbf{X}_n) = \text{out}(R \cup E \cup T \cup M \cup \mathbf{X}_n)$$

Figure 3.8

$$\geq (|R| + |U|) + |R| + |T| + |M| + 2n$$

$$\geq 4n + (|R| + |U|) + |M| - |T_1| \text{ (B6, B7)}$$

$$\geq 4n + |R| \text{ (B4)}$$

$$\geq 5n \text{ (as } |R| \geq n \text{ from (A2))}$$

Thus;

$$\mathbf{C}^m(T_3^n) = \mathbf{C}^m(S) \geq 2.5n - 5.5$$

and theorem follows. \square

Theorem(3.10) yields a lower bound on T_k^n when k is fixed. We now present a general lower bound on T_k^n , which gives larger bounds for $k = \Theta(n)$, $k \leq \lceil n/2 \rceil$. Thus:

$$\forall 3 \leq k \leq \lceil n/2 \rceil$$

$$C^m(T_k^n) \geq \max \{2n + 3k, 2.5n + 1.5k\} - c$$

Where c is a constant.

For the majority function, we deduce a lower bound of $3.5n$, slightly improving the $3n$ lower bound of Bloniarz (1979).

The approach is a generalisation of the standard inductive gate elimination argument. Three ideas are central to the proof method: extending the definition of "family of functions" as used in the inductive step; the notion of the "distance" of T_k^n from MAJ_n ; and the concept of a *reduction*. The concepts employed are similar to those employed in the $2.5n$ lower bound on combinational complexity from Stockmeyer (1977). So instead of considering a family of monotone functions $\{f_1, \dots, f_n, \dots\}$, in which for each n there is *at most one* n -input function, we consider families of *sets* of functions:

$$\{\{F_1\}, \{F_2\}, \dots, \{F_n\}, \dots\}$$

In this way each $f \in F_n$ is an n -input function. For the inductive step it is then sufficient to project onto a member of a smaller indexed set. The family we shall use is:

$$\bigcup_{n=2}^{\infty} \left\{ \bigcup_{k=2}^n \{T_k^n\} \right\}$$

Thus the n 'th member is the set:

$$\{T_2^n, T_3^n, \dots, T_{n-1}^n\}$$

The "distance" of T_k^n from majority is *related to* the minimum value of $|\pi|$, where π is the partial assignment such that:

$$(T_k^n)^{|\pi|} = MAJ_{n-|\pi|} \text{ and } n - |\pi| \text{ is even}$$

Using these concepts the lower bound proof divides into three parts: we first show how an arbitrary reduction may be used to reason about the size of monotone networks computing T_k^n ; then, assuming the correctness of a specific reduction, it is proved that a particular piecewise-linear function $\chi(n, k)$, gives lower bounds for T_k^n . The final stage is to verify the correctness of this reduction. This is done by a case analysis on the structure of optimal networks.

Definition 3.8: Define $\Delta(T_k^n)$ to be $n/2 - k$. Δ represents the "distance" of T_k^n from MAJ_n and may be negative and non-integral. •

Definition 3.9: Let S be a monotone network computing T_k^n . Let π be a partial assignment such that $S \xrightarrow{\pi} S'$, i.e $S^{|\pi|} = S'$, where S' computes T_k^{n-r} . The *descriptor* of π , $\delta(\pi)$, is a triple (r, s, t) where:

$$r = |\{ \text{Inputs of } S \text{ set by } \pi \}|$$

$$s = \Delta(T_k^{n-r})$$

$$t \leq |\{ \text{Gates deleted from } S \text{ by applying } \pi \}| \quad \bullet$$

Definition 3.10: An $\alpha\beta$ -reduction for T_k^n , is a set of q descriptor pairs, $\{ \langle a_i, b_i \rangle \}$ such that:

For any S computing T_k^n , $\exists \langle a_i, b_i \rangle$ and partial assignments π , π' applicable to S for which:

$$\delta(\pi) \in \{ a_i, b_i \} \tag{3.12}$$

$$\delta(\pi) = a_i \iff \delta(\pi') = b_i \quad (3.13)$$

$$\forall \langle a_i, b_i \rangle \quad 2\Delta(T_k^n) - (s_i + s'_i) = 0 \quad (3.14)$$

•

Lemma 3.12: (Dunne, 1984b) Let S compute T_k^n and let $\{\langle a_i, b_i \rangle\}$ be an $\alpha\beta$ -reduction for S . Let $\Delta(T_k^n) = s$. If there is a function $\chi(n, s) \rightarrow \mathbf{Q}^+$ such that:

$$\chi(n, s) \leq \max \begin{cases} \chi(n - r_i, s_i) + t_i \\ \chi(n - r'_i, s'_i) + t'_i \end{cases} \quad (3.15)$$

$\forall \langle a_i, b_i \rangle \equiv \langle (r_i, s_i, t_i), (r'_i, s'_i, t'_i) \rangle$ and

$$\chi(n, \Delta(T_1^n)) = \chi(n, \Delta(T_{n-1}^n)) = n - \text{Constant}$$

then $\mathbf{C}^m(T_k^n) \geq \chi(n, s)$.

Proof: By induction on n . For the inductive base the recurrence of (3.15) will terminate at $\chi(n, \Delta(T_1^n))$ or $\chi(n, \Delta(T_n^n))$. The conditions on χ yield the lower bound. For the inductive step assume $\forall n' < n$, $\forall k'$ that $\mathbf{C}^m(T_{k'}^{n'}) \geq \chi(n', s')$ where $\Delta(T_{k'}^{n'}) = s'$. and let S be a monotone network computing T_k^n . As $\{\langle a_i, b_i \rangle\}$ is an $\alpha\beta$ -reduction, there exist partial assignments π, π' , applicable to S , such that: $\langle \delta(\pi), \delta(\pi') \rangle = \langle a_i, b_i \rangle$ for some $1 \leq i \leq q$. Thus:

$$\mathbf{C}^{\mathbf{m}}(T_k^n) \geq \max \left\{ \begin{array}{l} \mathbf{C}^{\mathbf{m}}(T_{\binom{n-r_i}{2} \pm s_i}^{n-r_i}) + t_i \\ \mathbf{C}^{\mathbf{m}}(T_{\binom{n-r'_i}{2} \pm s'_i}^{n-r'_i}) + t'_i \end{array} \right.$$

By the inductive hypothesis:

$$\mathbf{C}^{\mathbf{m}}(T_k^n) \geq \max \left\{ \begin{array}{l} \chi(n - r_i, s_i) + t_i \\ \chi(n - r'_i, s'_i) + t'_i \end{array} \right.$$

But:

$$\chi(n, s) \leq \max \left\{ \begin{array}{l} \chi(n - r_i, s_i) + t_i \\ \chi(n - r'_i, s'_i) + t'_i \end{array} \right.$$

Hence: $\mathbf{C}^{\mathbf{m}}(T_k^n) \geq \chi(n, s)$ \square

Lemma(3.12) yields a recurrence expression for the monotone network complexity of T_k^n . We do not attempt to find a general solution to this, but illustrate that a particular $\chi(n, s)$ is given by a specified $\alpha\beta$ -reduction.

Lemma 3.13: (Dunne, 1984b) If:

$$AB = \{ \langle (1, s + 1/2, 4), (1, s - 1/2, 3) \rangle, \quad (3.16)$$

$$\langle (1, s + 1/2, 5), (1, s - 1/2, 2) \rangle \quad (3.17)$$

$$\langle (2, s + 1, 8), (2, s - 1, 6) \rangle \quad (3.18)$$

$$\langle (1, s + 1/2, 3), (1, s - 1/2, 4) \rangle \quad (3.19)$$

$$\langle (1, s + 1/2, 2), (1, s - 1/2, 5) \rangle \quad (3.20)$$

$$\langle (2, s + 1, 6), (2, s - 1, 8) \rangle \quad (3.21)$$

is an $\alpha\beta$ -reduction for every S computing $T_{n/2-s}^n$, then

$$\chi(n, s) = \begin{cases} 3.5n - |s| - c & 0 \leq |s| \leq 3/2 \\ 3.5n - 3|s| + 3 - c & |s| \geq 3/2 \end{cases}$$

satisfies:

$$\chi(n, s) \leq \max \begin{cases} \chi(n - r_i, s_i) + t_i \\ \chi(n - r'_i, s'_i) + t'_i \end{cases}$$

$$\forall \langle a_i, b_i \rangle \equiv \langle (r_i, s_i, t_i), (r'_i, s'_i, t'_i) \rangle \in AB$$

Proof: By inspection. \square

In terms of the usual form of inductive argument, $\chi(n, s)$ can be viewed as follows:

For any monotone network S_0 which realises T_k^n , one can find partial assignments $\pi_1, \pi_2, \dots, \pi_r$ such that:

$$(S_i)^{\pi_{i+1}} = S_{i+1} \quad \forall 0 \leq i < r$$

and the network S_r computes a threshold function which is "close to" majority. Then, for any T_k^n , close to majority, it is possible to choose partial assignments, π , which eliminate, on

average, 3.5 gates and such that $(T_k^n)^{\lceil \pi \rceil}$ is also close to majority.

We observe that the $\alpha\beta$ -reduction AB , can be similarly interpreted, for a number of different $\chi(n, s)$. One such interpretation is outlined below.

It may be noted that in some $\langle a_i, b_i \rangle$:

$$\chi(n, s) \geq \min \begin{cases} \chi(n - r_i, s_i) + t_i \\ \chi(n - r'_i, s'_i) + t'_i \end{cases}$$

e.g $\chi(n, 1/2) > \chi(n - 2, -1/2) + 6$

This imposes a strategy in inductively eliminating gates from S computing $T_{n/2-s}^n$, in that for those $\langle a_i, b_i \rangle$ having this property the step which reduces to:

$$\chi(n, s) \leq \max \begin{cases} \chi(n - r_i, s_i) + t_i \\ \chi(n - r'_i, s'_i) + t'_i \end{cases}$$

must be applied.

Theorem 3.11: Let S be any optimal network computing T_k^n for $1 < k < n$. Then AB is an $\alpha\beta$ -reduction for S .

Proof: The proof is by a case analysis on the fanout of inputs. For reasons of space the details are omitted. A complete description is given in Dunne (1984b). \square

Corollary 3.6: $\forall k \ 3 \leq k \leq \lceil n/2 \rceil$

$$\mathbf{C}^{\mathbf{m}}(T_k^n) \geq 2n + 3k - c \quad c \in \mathbf{Q}^+$$

Proof: Let $k = n/2 - s$, $s \in \mathbf{Q}^+$. By Lemma(3.13) and Theorem(3.11)

$$\mathbf{C}^{\mathbf{m}}(T_k^n) = \mathbf{C}^{\mathbf{m}}(T_{n/2-s}^n) \geq 3.5n - 3s + 3 - c'$$

However: $s = n/2 - k$, thus

$$\begin{aligned} \mathbf{C}^{\mathbf{m}}(T_k^n) &\geq 3.5n - 3(n/2 - k) - c' \\ &\geq 2n + 3k - c \quad \square \end{aligned}$$

Corollary 3.7:

$$\mathbf{C}^{\mathbf{m}}(T_k^n) \geq 4(k-3) + \mathbf{C}^{\mathbf{m}}(T_3^{n-k+3}) - c$$

Proof: (Outline) The $\alpha\beta$ -reduction AB may be interpreted by saying:

"For any monotone network S computing T_k^n , \exists some input x_i and some constant $c \in \{0, 1\}$ such that setting $x_i = c$ eliminates at least 4 gates."

Choosing a suitable $\chi(n, \Delta(T_k^n))$ leads to the theorem. \square

Corollary 3.8: If $\mathbf{C}^{\mathbf{m}}(T_3^n) = (2 + \lambda)n - c$ then $\forall 3 \leq k \leq \lceil n/2 \rceil$

$$\begin{aligned} \mathbf{C}^{\mathbf{m}}(T_k^n) &\geq \max \begin{cases} 2n + 3k - c_0 \\ (2 + \lambda)n + (2 - \lambda)k - c_1 \end{cases} \\ &\geq \begin{cases} 2n + 3k - c_0 & k \geq \frac{\lambda n}{\lambda + 1} \\ (2 + \lambda)n + (2 - \lambda)k - c_1 & k \leq \frac{\lambda n}{\lambda + 1} \end{cases} \end{aligned}$$

\square

Theorem 3.12: (Dunne, 1984b)

$$\mathbf{C}^m(T_k^n) \geq \begin{cases} 2n + 3k - c_0 & k \geq n/3 \\ 2.5n + 1.5k - c_1 & k \leq n/3 \end{cases}$$

Proof: From Theorem(3.10) $\mathbf{C}^m(T_3^n) \geq 2.5n - 5.5$ and the theorem follows from Corollary(3.8) with $\lambda=1/2$. \square

Corollary(3.8) implies that improved lower bounds on T_3^n or any T_k^n with k fixed, would lead to consequent improvements in Theorem(3.12). In particular a $3n$ lower bound on T_3^n would immediately give the $3.5n$ lower bound on MAJ_n .

The final linear lower bound presented is from Tiekenheinrich (1984) and is notable for the absence of the cumbersome technical complications of the preceding methods.

Lemma 3.14: Any monotone network realising $T_2^n(\mathbf{X}_n)$ contains at least $2n - 4$ \vee -gates.

Proof: By induction on $n \geq 2$. Since the base is trivial assume the lemma is true for all values $< n$ and let S be a monotone network realising $T_2^n(\mathbf{X}_n)$ at some node t . Let g be a gate of S whose distance from t is maximal. The inputs of g must be distinct inputs x_i and x_j of S . From Thm(1.15), $\phi(x_i) \geq 2$. If x_i enters at least 2 \vee -gates then setting $x_i = 0$ proves the result via the inductive hypothesis. If x_i enters only one \vee -gate, h_1 say and an \wedge -gate h_2 then proceed as follows. Let u be the first \vee -gate encountered on some path from h_2 to t . u must exist otherwise setting x_i to 0 makes t equal 0. If $u \neq h_1$ then setting $x_i = 0$ eliminates 2 \vee -gates from S . If $u = h_1$, then there must be some \vee -gate on any path from h_1 to t . The first of these can be eliminated by setting x_i to 0. The case when x_i enters no \vee -gates,

and thus at least 2 \wedge -gates, is dealt with by a similar argument. \square

Theorem 3.13: (Tiekenhienrich, 1984) Let $f(\mathbf{X}_n, z) \in M_{n+1}$ be defined by

$$f(\mathbf{X}_n, z) = z \wedge T_2^n(\mathbf{X}_n) \vee T_{n-1}^n(\mathbf{X}_n)$$

then

$$\mathbf{C}^m(f) \geq 4n - 8$$

Proof: Let S be an optimal monotone network realising $f(\mathbf{X}_n, z)$. Since $f|_{z:=1} = T_2^n(\mathbf{X}_n)$, from Lemma(3.14), S must contain at least $2n - 4$ \vee -gates. Since $f|_{z:=0} = T_{n-1}^n(\mathbf{X}_n) = \tilde{T}_2^n(\mathbf{X}_n)$, by duality S must contain also at least $2n - 4$ \wedge -gates. This is a total of $4n - 8$ distinct gates and so the theorem follows. \square

The lower bound on T_k^n proved above, is possibly sub-optimal. This section concludes with a monotone network construction for T_k^n . This gives the upper bound $\mathbf{C}^m(T_k^n) \leq kn + o(n)$ for k fixed. To prove the upper bound the following combinatorial result is required.

Fact 3.6: Let $\mathbf{y}_i = \langle y_{i_1}, y_{i_2}, \dots, y_{i_k} \rangle \in \mathbf{N}^k$ (where $k \geq 2$) and $\Pi_q: \mathbf{N}^k \rightarrow \mathbf{N}^k$ be the projection which sets the y_{i_q} position of \mathbf{y}_i to 1. Finally let $COVER_k$ be a predicate defined on sets of k -tuples $\mathbf{Y} = (\mathbf{y}_1, \dots, \mathbf{y}_s)$ by:

$$COVER_k(\mathbf{y}_1, \dots, \mathbf{y}_s) = \begin{cases} 1 & \text{if } \forall 1 \leq q \leq k, \exists \mathbf{y}_i^q, \mathbf{y}_j^q \in \mathbf{Y} \\ & \text{such that } \Pi_q(\mathbf{y}_i^q) = \Pi_q(\mathbf{y}_j^q) \text{ and } i \neq j \\ 0 & \text{otherwise} \end{cases}$$

Then $\min_{\mathbf{Y} \subset \mathbf{N}^k} \{|\mathbf{Y}| : COVER_k(\mathbf{Y}) = 1\} = k + 1$

Proof: The upper bound is elementary. The lower bound is proved by induction on $k \geq 2$. The base $k = 2$ is immediate, so we assume the lower bound holds for all values less than k . Let $\mathbf{Y} = \{\mathbf{y}_1, \dots, \mathbf{y}_s\}$ be any set of k -tuples such that $COVER_k(\mathbf{Y}) = 1$. Without loss of generality it may be assumed that $\Pi_1(\mathbf{y}_1) = \Pi_1(\mathbf{y}_2)$. Thus, as $COVER_k(\mathbf{Y}) = 1$, the set of $(s - 1)$ $(k - 1)$ -tuples

$$\{ \langle y_{1_2}, \dots, y_{1_k} \rangle \} \cup \bigcup_{i=3}^s \{ \langle y_{i_2}, \dots, y_{i_k} \rangle \}$$

must satisfy $COVER_{k-1}$. By the inductive hypothesis $s - 1 \geq k \Leftrightarrow s \geq k + 1$. The lower bound follows. \square

Theorem 3.14: For any fixed k $C^m(T_k^n(\mathbf{X}_n)) \leq kn + o(n)$.

Proof: For ease of exposition, suppose $n = p^k$ for some positive integer p . It is easy to see how to amend the construction below if n is not of this form. Let:

$$\mathbf{X}_n = \bigcup_{\substack{1 \leq r_1 \leq p \\ 1 \leq r_2 \leq p \\ \dots \\ \dots \\ 1 \leq r_k \leq p}} \{ x_{r_1 r_2 \dots r_k} \}$$

To avoid a plethora of subscripts $\langle r_1, \dots, r_k \rangle$ will denote $x_{r_1 \dots r_k}$. It will be convenient to consider the elements of $\{1, 2, \dots, p\}^{k-1}$ arranged in lexicographic order. Thus $\mathbf{r}_i = \langle r_i^1, r_i^2, \dots, r_i^{k-1} \rangle$ is the i 'th element. e.g $\mathbf{r}_1 = \langle 1, 1, 1, \dots, 1 \rangle$

The q -partition of \mathbf{X}_n is constructed as follows.

T1) \mathbf{X}_n is partitioned into p^{k-1} blocks, B_i^q , where $1 \leq i \leq p^{k-1}$. Each block contains p elements of \mathbf{X}_n .

T2) The particular elements of \mathbf{X}_n in a block B_i^q are given by:

$$B_i^q = \bigcup_{j=1}^p \{ \langle r_i^1, \dots, r_i^{q-1}, j, r_i^{q+1}, \dots, r_i^k \rangle \}$$

where $\langle r_i^1, \dots, r_i^k \rangle$ is the i 'th element of $\{1, 2, \dots, p\}^{k-1}$ in the ordering described above.

The q -partition of \mathbf{X}_n thus consists of p^{k-1} blocks each block being defined by a distinct $(k-1)$ -tuple.

Clearly there are k possible q -partitions of \mathbf{X}_n . We claim that:

$$T_k^{p^k}(\mathbf{X}_n) = \bigvee_{q=1}^k T_k^{p^{k-1}}(T_1^p(B_1^q), \dots, T_1^p(B_{p^{k-1}}^q)) \quad (3.22)$$

If this assertion holds, it gives rise to a recursive construction for a monotone network computing T_k^n . Solving the underlying recurrence relation yields the upper bound stated. We justify this assertion as follows. First observe that if fewer than k element of \mathbf{X}_n are assigned the value 1 then the right-hand side of (3.22) is 0. Since the RHS is clearly monotone it is sufficient to prove that it attains the value 1 whenever exactly k members of \mathbf{X}_n are 1. Consider any assignment to \mathbf{X}_n for which exactly k variables are set to 1. Let:

$$\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_k\}$$

$$= \{ \langle y_{1_1}, y_{1_2}, \dots, y_{1_k} \rangle, \dots, \langle y_{k_1}, y_{k_2}, \dots, y_{k_k} \rangle \}$$

be the k variables of \mathbf{X}_n which are fixed to 1. From Fact(3.6), since $|\mathbf{Y}| < k+1$, $COVER_k(\mathbf{Y}) = 0$. It follows that there exists some s (with $1 \leq s \leq k$) such that:

$$\{ \langle y_{1_1}, \dots, y_{1_{s-1}}, y_{1_{s+1}}, \dots, y_{1_k} \rangle, \dots, \langle y_{k_1}, \dots, y_{k_{s-1}}, y_{k_{s+1}}, \dots, y_{k_k} \rangle \}$$

are distinct $(k-1)$ -tuples in $\{1, 2, \dots, p\}^{k-1}$. Therefore by the definition

of q -partition:

$$\mathbf{y}_i \in B_i^s \wedge \mathbf{y}_j \in B_i^s \iff i = j$$

Thus no two \mathbf{y}_i 's (i.e variables of \mathbf{X}_n which are set to 1) are in the same block of the s -partition of \mathbf{X}_n . So:

$$T_k^{p^{k-1}} (T_1^p(B_1^s), T_1^p(B_2^s), \dots, T_1^p(B_{p^{k-1}}^s)) = 1$$

and therefore the RHS of (3.22) is 1. \square

3.5) Superpolynomial Lower Bounds on Single Output Functions

In this section we give a detailed technical description of two techniques for deriving large lower bounds on the monotone complexity of single output functions; that of Razborov as enhanced by Alon and Boppana (1986), and that of Andreev. In the remainder of this section we give some basic definitions and notation. In Section(3.5.1) the Razborov, Alon and Boppana results are presented. Section(3.5.2) describes the conceptually simpler approach employed in Andreev (1985). We conclude with a discussion and comparison of the two methods.

It will sometimes be convenient to regard a monom as the set of variables defining it, as well as a function. In this way we may write $m_1 \cap m_2$ and $m_1 \cup m_2$ instead of $var(m_1) \cap var(m_2)$ and $var(m_1) \cup var(m_2)$.

In the next section we will be interested in the following monotone Boolean functions.

Recall that $\mathbf{X}_n^U = \{x_{ij} : 1 \leq i < j \leq n\}$ denotes a set of $n/2$ Boolean variables. $G(\mathbf{X}_n^U)$ is a function from assignments, α , to \mathbf{X}_n^U onto n -vertex undirected graphs in which $G(\alpha)$ contains an edge $\{i, j\}$ if and only if $x_{ij} = 1$ in the assignment α to \mathbf{X}_n^U . For $1 \leq k \leq n$ the function k -clique(\mathbf{X}_n^U) takes the value 1 if and only if $G(\mathbf{X}_n^U)$ contains a k -clique, i.e a set $\{v_1, \dots, v_k\} \subseteq \{1, \dots, n\}$ of vertices such that for all $1 \leq i \neq j \leq k$, $\{v_i, v_j\}$ is an edge of $G(\mathbf{X}_n^U)$.

$\mathbf{X}_{n,n} = \{x_{i,j} : 1 \leq i, j \leq n\}$ denotes a set of n^2 Boolean variables. $B(\mathbf{X}_{n,n})$ is a mapping from assignments to $\mathbf{X}_{n,n}$ onto $2n$ -vertex bipartite graphs. $B(\alpha)$ is a bipartite graph over two disjoint sets V and W of vertices ($|V| = |W| = n$) in which there is an edge between $v_i \in V$ and $W_j \in W$ if and only if $x_{i,j} = 1$ under α . $PM(\mathbf{X}_{n,n})$ is the monotone Boolean function which takes the value 1 if and only if the

bipartite graph $B(\mathbf{X}_{n,n})$ contains a *perfect matching*, i.e there is a $2n$ -vertex subgraph (factor) of $B(\mathbf{X}_{n,n})$ in which every vertex is the end-point of exactly one edge. $PM(\mathbf{X}_{n,n})$ may be expressed as

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

Notation: For any finite set F , 2^F denotes the set of all subsets of F and $P_s(F)$ the set of all subsets of F having cardinality $\leq s$. We use $2^{\mathbf{X}_n}$ to denote the set $\mathbf{I}(1)$ and \emptyset to denote $\mathbf{I}(0)$.

3.5.1) The Lattice Method of Razborov, Alon and Boppana

This technique is presented in Razborov (1985a,b) and was improved by Alon and Boppana (1986). Our description below combines results from all of these papers. The method divides naturally into three stages, of which only the last is heavily dependent on the monotone function considered. At the core of the approach is a novel interpretation of monotone Boolean networks as a particular type of combinatorial structure called a regular lattice.

Definition 3.11: A *regular lattice*, \mathbf{M} , is a lattice whose elements are subsets of $2^{\mathbf{X}_n}$ and which satisfies,

R1) $\{\mathbf{I}(x_1), \dots, \mathbf{I}(x_n), \mathbf{I}(0), \mathbf{I}(1)\} \subseteq \mathbf{M}$

R2) Elements of \mathbf{M} are ordered by set containment \subseteq .

\sqcap and \sqcup denote the usual lattice operations *meet* and *join*. Given A, B elements of a regular lattice \mathbf{M} these operations satisfy,

$$A \cup B \subseteq A \sqcup B \quad ; \quad A \sqcap B \subseteq A \cap B \quad \bullet$$

In general both these containments will be strict. This motivates the ideas of *surplus* and *deficiency*. For A, B as before these are given

respectively by

$$\begin{aligned}\delta_+(A, B) &= (A \cap B) - (A \sqcap B) \\ \delta_-(A, B) &= (A \sqcup B) - (A \cup B)\end{aligned}\tag{3.23}$$

Finally in order to model monotone networks by regular lattices some concept of the complexity of a function with respect to any such lattice is needed.

Definition 3.12: Let $f(\mathbf{X}_n) \in M_n$ and \mathbf{M} be any regular lattice. The *distance* of f in \mathbf{M} , denoted $\rho(f, \mathbf{M})$, is the least t such that,

\exists t pairs of lattice elements $\langle A_i, B_i \rangle$ and an element D of \mathbf{M} for which

$$D \subseteq \mathbf{I}(f) \cup \bigcup_{i=1}^t \delta_-(A_i, B_i)\tag{3.24}$$

$$\mathbf{I}(f) \subseteq D \cup \bigcup_{i=1}^t \delta_+(A_i, B_i)\tag{3.25}$$

•

Using these ideas the three sections of the technique consist of

- S1) Showing that for all $f \in M_n$ and all regular lattices \mathbf{M} , $\mathbf{C}^m(f) \geq \rho(f, \mathbf{M})$.
- S2) Constructing a particular class of regular lattices $CLOSED(f)$.
- S3) Proving a lower bound on the monotone complexity of some specific functions, f , by obtaining a lower bound on the distance of f in $CLOSED(f)$. Here the method relies on the structure of the given f in order to derive large bounds.

3.5.1.1. Distance in Regular Lattices and Monotone Complexity

Lemma 3.15: $\forall f(\mathbf{X}_n) \in M_n, \forall$ regular lattices \mathbf{M} ;

$$\mathbf{C}^{\mathbf{m}}(f) \geq \rho(f, \mathbf{M})$$

Proof: Let S be any monotone network realising f with $\mathbf{C}^{\mathbf{m}}(S) = t$. Number the gates of S in topological order. Let l_i, r_i denote the functions computed by the left (resp. right) input nodes for the gate numbered i .

With each node i of S we associate an element $\lambda(v)$ of \mathbf{M} as follows,

$$\lambda(v) = \begin{cases} \mathbf{I}(x_j) & \text{if } v \text{ is the input } e_j \\ \mathbf{I}(0) & \text{if } v \text{ is an input labelled with } 0 \\ \mathbf{I}(1) & \text{if } v \text{ is an input labelled with } 1 \\ \lambda(v_L) \sqcap \lambda(v_R) & \text{if } op(v) = \wedge \\ \lambda(v_L) \sqcup \lambda(v_R) & \text{if } op(v) = \vee \end{cases}$$

We claim that choosing the t pairs $\langle A_i, B_i \rangle$ by $A_i = \lambda(i_L)$, $B_i = \lambda(i_R)$, where i is the gate numbered i by the topological ordering of S , (thus $1 \leq i \leq t$); and choosing $D = \lambda(t)$ satisfies relations (3.24) and (3.25) above. This is easily shown by induction on $t \geq 0$. If $t = 0$ then f is either a constant function or the variable x_i . In any case D is defined as $\mathbf{I}(f)$ and so the inductive base is trivial.

Now suppose the assertion holds for all values $\leq t - 1$. We shall prove it holds for t also. Since $t \geq 1$, S contains at least one gate. Consider the output gate of S , which must be labelled t in the topological ordering. By definition we have that $f = l_t op(t) r_t$. By the inductive hypothesis, since both l_t and r_t are computed using at most $t - 1$ gates in S , we have:

$$A_t \subseteq \mathbf{I}(l_t) \cup \bigcup_{i=1}^{t-1} \delta_-(A_i, B_i) \quad (3.26)$$

$$B_t \subseteq \mathbf{I}(r_t) \cup \bigcup_{i=1}^{t-1} \delta_-(A_i, B_i) \quad (3.27)$$

$$\mathbf{I}(l_t) \subseteq A_t \cup \bigcup_{i=1}^{t-1} \delta_+(A_i, B_i) \quad (3.28)$$

$$\mathbf{I}(r_t) \subseteq B_t \cup \bigcup_{i=1}^{t-1} \delta_+(A_i, B_i) \quad (3.29)$$

First suppose that $op(t) = \vee$. In this case,

$$D = \lambda(t) = A_t \sqcup B_t$$

and so using (3.26) and (3.27);

$$\begin{aligned} D &= A_t \cup B_t \cup \delta_-(A_t, B_t) \\ &\subseteq \mathbf{I}(l_t) \cup \mathbf{I}(r_t) \cup \bigcup_{i=1}^t \delta_-(A_i, B_i) \\ &= \mathbf{I}(f) \cup \bigcup_{i=1}^t \delta_-(A_i, B_i) \end{aligned}$$

So (3.24) holds in this case. Similarly using (3.28) and (3.29);

$$\begin{aligned} \mathbf{I}(f) &= \mathbf{I}(l_t) \cup \mathbf{I}(r_t) \\ &\subseteq A_t \cup B_t \cup \bigcup_{i=1}^{t-1} \delta_+(A_i, B_i) \end{aligned}$$

$$\begin{aligned} &\subseteq (A_t \sqcup B_t) \cup \bigcup_{i=1}^{t-1} \delta_+(A_i, B_i) \\ &\subseteq D \cup \bigcup_{i=1}^t \delta_+(A_i, B_i) \end{aligned}$$

So that (3.25) holds. The case $op(t) = \wedge$ can be shown by a similar argument. This completes the inductive argument and hence, $\mathbf{C}^m(f) \geq \rho(f, \mathbf{M})$. \square

3.5.1.2. The Class of Regular Lattices $CLOSED(f)$

Lemma(3.15) demonstrates that in order to prove lower bounds on monotone complexity it is sufficient to prove lower bounds on distance in regular lattices. There are infinitely many possible choices of regular lattice. A lattice with too few elements e.g containing only $\mathbf{I}(x_i)$, \emptyset and $2^{\mathbf{X}_n}$, would have insufficient structure to allow large bounds to be easily derived. On the other hand, using a lattice with too many elements e.g containing every subset of $2^{\mathbf{X}_n}$, it would only be possible to derive trivial lower bounds on distance, in the cited example every function has distance 0 only.

Razborov defined a class of regular lattices based on the prime implicant structure of any given monotone function and a novel closure relation. Unfortunately no motivation for the chosen representation is given and the reader should note that much of the development below is non-intuitive.

Let $f(\mathbf{X}_n) \in M_n$ and recall that $\mathbf{PI}(f)$ is the set of prime implicants of f . Below $r \geq 2$ and $s \geq 1$ are natural numbers.

Definition 3.13: For $f \in M_n$, $\mathbf{U}(f) \subseteq P_s(\mathbf{X}_n)$ is the set,

$$\mathbf{U}(f) = \{m : |m| \leq s \text{ and } \exists p \in \mathbf{PI}(f) \text{ such that } p \leq m\}$$

Here we are regarding m both as a function and as a set of variables. •

Informally \mathbf{U} is the set of monoms containing at most s variables and which are shortenings of prime implicants of $f(\mathbf{X}_n)$.

The relation \vdash (read "yields") is a subset of $\mathbf{U}(f)^r \times \mathbf{U}(f)$ defined by saying $\langle E_1, \dots, E_r \rangle \vdash E_0$ (where $E_i \in \mathbf{U}(f)$, $0 \leq i \leq r$) if and only if

$$\bigcup_{1 \leq i < j \leq r} E_i \cap E_j \subseteq E_0 \tag{3.30}$$

If $U_1 \subseteq \mathbf{U}$ and $E \in \mathbf{U}$ we write $U_1 \vdash E$ if there are sets E_1, \dots, E_r (not necessarily distinct) in U_1 such that $\langle E_1, \dots, E_r \rangle \vdash E$. A subset U_1 of \mathbf{U} is said to be *closed* if $\forall E \in \mathbf{U}(f) (U_1 \vdash E \Leftrightarrow E \in U_1)$. We denote by \bar{U}_1 the smallest cardinality closed subset of $\mathbf{U}(f)$ which contains U_1 , thus if U_1 is closed then $\bar{U}_1 = U_1$. This is called the *closure* of U_1 .

Finally, for any $E \in \mathbf{U}(f)$ the *cover* of E , denoted $\lceil E \rceil$ is the set,

$$\lceil E \rceil = \{ F : F \supseteq E \}$$

For a subset U_1 of $\mathbf{U}(f)$, this is naturally extended by defining, the set $\lceil U_1 \rceil$ as $\bigcup_{E \in U_1} \lceil E \rceil$. •

Definition 3.14: Let $f(\mathbf{X}_n) \in M_n$ the lattice $CLOSED(f)$ is the lattice which contains exactly the elements,

$$\{ \lceil U_1 \rceil : U_1 \subseteq \mathbf{U}(f) \text{ and } U_1 \text{ is closed} \} \quad \bullet$$

$CLOSED(f)$ is the lattice used to derive the lower bounds

established below.^{g)}

Lemma 3.16: For all $f \in M_n$, $CLOSED(f)$ is a regular lattice, whose \sqcap and \sqcup operations satisfy for all $\lceil A \rceil, \lceil B \rceil \in CLOSED(f)$,

$$\lceil A \rceil \sqcap \lceil B \rceil = \lceil A \cap B \rceil$$

$$\lceil A \rceil \sqcup \lceil B \rceil = \lceil \overline{A \cup B} \rceil$$

Proof: It is clear that $CLOSED(f)$ contains $\mathbf{I}(x_i) = \lceil \overline{\{x_i\}} \rceil$ for all $1 \leq i \leq n$ and also $\mathbf{I}(0) = \lceil \emptyset \rceil$ and $\mathbf{I}(1) = \lceil \mathbf{U}(f) \rceil$. For the second part of the lemma consider

$$SUB(f) = \{ C : C \text{ is a closed subset of } \mathbf{U}(f) \}$$

Obviously $SUB(f)$ under the ordering \subseteq forms a lattice with operations $inf(A, B) \equiv A \cap B$ and $sup(A, B) \equiv \overline{A \cup B}$. The cover operation $\lceil \dots \rceil$ defines a mapping from $SUB(f) \rightarrow CLOSED(f)$ which is an order preserving lattice homomorphism, i.e

$$A \subseteq B \Leftrightarrow \lceil A \rceil \subseteq \lceil B \rceil$$

The lemma will follow if $\lceil \dots \rceil$ is actually an *isomorphic mapping*. Thus if for any closed subsets A, B of $\mathbf{U}(f)$ we have

$$\lceil A \rceil \subseteq \lceil B \rceil \Leftrightarrow A \subseteq B$$

So suppose A, B are two closed subsets of $\mathbf{U}(f)$ with $\lceil A \rceil \subseteq \lceil B \rceil$. Let $E \in A$. Since

$$E \in A \subseteq \lceil A \rceil \subseteq \lceil B \rceil$$

g) For the lower bound on k -clique this is not strictly true. Instead of using \mathbf{U} as the basis for defining "closure", \vdash and "covers" a particular subset of \mathbf{U} is used. The combinatorial results proved subsequently all hold for the lattice structure that arises. This will be clear when the actual form used is defined later.

it follows from the definition of $\lceil \dots \rceil$ that there is some $F \in B$ such that $F \subseteq E$, we therefore have from (3.30) that

$$F, F, \dots, F \vdash E$$

and $E \in B$ since this is a closed subset of \mathbf{U} . So we have established that $A \subseteq B$ proving the lemma. \square

A set $C \subseteq 2^{\mathbf{X}_n}$ (i.e set of monoms or of variable sets) is said to be *independent* if for each distinct $A, B \in C$: $A \not\subseteq B$ and $B \not\subseteq A$.

The lower bound proofs require upper bounds on two measures to be established:

UPB1) For any closed set $W \subseteq \mathbf{U}(f)$ the value of $|base_k(W)|$, where $base_k(W)$ is

$$\{ E \in W : |E| \leq k \text{ and } \forall F \in W F \subseteq E \Leftrightarrow F = E \}$$

$base_k(W)$ consists of the minimal (w.r.t \subseteq) sets in W of cardinality at most k .

UPB2) For any set $W \subseteq \mathbf{U}(f)$ the value of $|\bar{W} - W|$, i.e the number of sets added to W to render it closed.

The main contribution made by Alon and Boppana was in improving the upper bounds on these quantities originally obtained in Razborov (1985a,b). In practice only the improvement of (UPB1) led to larger lower bounds, although in principle that made to (UPB2) could also yield better results. At present no examples were this is so are known.

We tackle the problem posed by (UPB1) in two stages: first it is shown that for any independent subset, W , of $P_k(\mathbf{X}_n)$ for which there does not exist any $r + 1$ -tuple

$$\langle E_0, E_1, \dots, E_r \rangle \in W^{r+1}$$

with $\bigcup_{1 \leq i < j \leq r} E_i \cap E_j \subset E_0$, such a set contains at most $(r-1)^k$ members. Note that the containment is *strict*. Using this it is easy to derive an upper bound on

$$I_k = \max \{ |base_k(W)| : W \subseteq \mathbf{U}(f) \text{ and } W \text{ is closed } \}$$

Following Alon and Boppana, we say that an independent subset W of $P_k(\mathbf{X}_n)$ is *r-stable* if there does not exist any $r+1$ -tuple in W^{r+1} with the property described in the preceding paragraph.^{h)}

Lemma 3.17: (Alon and Boppana, 1986) improving Razborov (1985a) Let $W \subseteq P_k(\mathbf{X}_n)$ be independent. If W is *r-stable* then $|W| \leq (r-1)^k$.

Proof: By induction on $r \geq 2$. For the inductive base, $r=2$ let $W \subseteq P_k(\mathbf{X}_n)$ be independent and 2-stable. If $|W| \geq 2$ then W contains sets W_1 and W_2 . But $W_1 \cap W_2 \subset W_1 \in W$ which contradicts the assumption of 2-stability. Thus $|W| = 1$ proving the inductive base.

For the inductive hypothesis assume the lemma holds for all values $\leq r-1$ and let $W \subseteq P_k(\mathbf{X}_n)$ be independent and *r-stable*. We shall show that $|W| \leq (r-1)^k$. Choose any $V \in W$ and for each $C \subseteq V$ define a set $W_C \subseteq P_{k-|C|}(\mathbf{X}_n)$ by

$$W_C = \{ F - C : F \in W \text{ and } F \cap V = C \}$$

Clearly W_C is independent. W_C is also $(r-1)$ -stable. To see this suppose the contrary. Then there is some r -tuple $\langle E_0, E_1, \dots, E_{r-1} \rangle \in W_C^r$ such that

$$\bigcup_{1 \leq i < j \leq r-1} E_i \cap E_j \subset E_0$$

But this implies that,

^{h)} Alon and Boppana (1986) actually uses the term "has Property $P(r, k)$ ".

$$\bigcup_{1 \leq i < j \leq r-1} (E_i \cup C) \cap (E_j \cup C) \subset E_0 \cup C$$

and since $E_i \in W_C$, by definition we have that

$$\bigcup_{1 \leq i \leq r-1} (E_i \cup C) \cap V = C$$

So if we choose $F_i = E_i \cup C$, for each $0 \leq i \leq r-1$ and set $F_r = V$ then $\langle F_0, F_1, \dots, F_r \rangle \in W^{r+1}$ and from the previous argument,

$$\bigcup_{1 \leq i < j \leq r} F_i \cap F_j \subset F_0$$

contradicting the r -stability of W . It follows that W_C is $(r-1)$ -stable.

By the inductive hypothesis, since $W_C \subseteq P_{k-|C|}(\mathbf{X}_n)$, this gives $|W_C| \leq (r-2)^{k-|C|}$. Now,

$$\begin{aligned} |W| &= \sum_{C \subseteq V} |W_C| \leq \sum_{C \subseteq V} (r-2)^{k-|C|} \\ &= \sum_{i=0}^{|V|} \binom{|V|}{i} (r-2)^{|V|-i} \\ &\leq \sum_{i=0}^k \binom{k}{i} (r-2)^{k-i} = (r-1)^k \end{aligned}$$

By the Binomial theorem. This completes the induction, proving the lemma. \square

Corollary 3.9: $I_k \leq (r-1)^k$

Proof: (Razborov, 1985b) Let W be any closed subset of $\mathbf{U}(f)$. It is sufficient to show that $|base_k(W)| \leq (r-1)^k$. Clearly $base_k(W) \subseteq P_k(\mathbf{X}_n)$ and is independent. Suppose $base_k(W)$ is not r -stable. As before we can find

$\langle E_0, E_1, \dots, E_r \rangle \in \text{base}_k(W)^{r+1} \subseteq W^{r+1}$ such that,

$$\bigcup_{1 \leq i < j \leq r} E_i \cap E_j = E \subset E_0$$

Hence $\text{base}_k(W) \Vdash E \Leftrightarrow W \Vdash E \Leftrightarrow E \in W$ by closure. But $E \in W$ contradicts $E_0 \in \text{base}_k(W)$ since $E \subset E_0$. This contradiction shows that $\text{base}_k(W)$ is r -stable, thus from Lemma(3.17) $|\text{base}_k(W)| \leq (r-1)^k$ as claimed. \square

We now turn to the problem posed in (UPB2), that of bounding the number of sets needed to produce \bar{W} from W . In fact all that is needed for subsequent development in the lower bound proofs is an upper bound on the number of iterations of the following algorithm to produce \bar{W} .

Input: A, B where $A, B \subseteq \mathbf{U}(f)$ and closed

Output: $\bar{W} = \overline{(A \cup B)}$

Method: (Alon and Boppana, 1986)

$W_0 := A \cup B$

while $W_i \neq \bar{W}$ **do**

$V_{i+1} := E \in \text{base}_s(\{F \notin W_i : W_i \Vdash F\})$

$W_{i+1} := W_i \cup (\lceil V_{i+1} \rceil \cap P_s(\mathbf{X}_n))$

$i := i + 1$

od

The Closure Algorithm

Let p be the maximal number of iterations of this algorithm, and $\langle V_1, V_2, \dots, V_p \rangle$ the sequence of (minimal) sets whose cover up to sets of cardinality s is added at each stage. We wish to derive an upper bound on p . Now since $\mathbf{U}(f)$ is finite and closed it is obvious that $p \leq |\mathbf{U}(f)|$ and this is the measure used in (Razborov, 1985a,b).

As we remarked previously even this crude estimate is adequate to derive the exponential bounds obtained in Alon and Boppana (1986). However the improved bound derived by Alon and Boppana may yet be of value for deriving further results.

Lemma 3.18: p the number of iterations of the Closure Algorithm is $\leq 2r^s$.

Proof: Let $S = \langle V_1, \dots, V_p \rangle$ be the sequence of minimal sets added by the closure algorithm. This sequence has the following property:

Each V_i has cardinality $\leq s$ and there do not exist $i_1 \leq i_2 \leq \dots \leq i_r < i_{r+1}$ for which

$$\langle V_{i_1}, \dots, V_{i_r} \rangle \vdash U \subset V_{i_{r+1}}$$

We say that any sequence of distinct sets $\langle C_1, \dots, C_q \rangle$ which satisfy this have *Property T*(r,s).

Now suppose S is as above but that S does not have property T . We claim that then $V_{i_{r+1}}$ would not be the set added at the i_{r+1} iteration. This is because $V_{i_{r+1}}$ is supposed to be a minimal set which is not in $W_{i_{r+1}-1}$ but such that $W_{i_{r+1}-1}$ yields $V_{i_{r+1}}$. Now if

$$\langle V_{i_1}, V_{i_2}, \dots, V_{i_r} \rangle \vdash U \subset V_{i_{r+1}}$$

i.e T does not hold, then $V_{i_{r+1}}$ could only be an appropriate set to consider if $U \in W_{i_{r+1}-1}$. This can be true only if $U \in A \cup B$ or $U \in \lceil V_j \rceil \cap P_s(\mathbf{X}_n)$ for some $j < i_{r+1}$. In the former case we have, without loss of generality, $U \in A$ so $\lceil U \rceil \cap P_s(\mathbf{X}_n) \subseteq A$ (since A is closed) hence, because $U \subset V_{i_{r+1}} \in P_s(\mathbf{X}_n)$, $V_{i_{r+1}} \in A$. This contradicts $V_{i_{r+1}}$ occurring in the sequence of sets S . The latter case is even easier to dismiss for then $\lceil U \rceil \cap P_s(\mathbf{X}_n) \subseteq \lceil V_j \rceil \cap P_s(\mathbf{X}_n)$ thus by the previous argument $V_{i_{r+1}} \in \lceil V_j \rceil \cap P_s(\mathbf{X}_n)$ and is again

unsuitable.ⁱ⁾

So $\langle V_1, \dots, V_p \rangle$ has property T . We claim that for all $r \geq 1$, $s \geq 0$ any sequence of distinct sets $\langle C_1, \dots, C_q \rangle$ with property T must have $q \leq 2r^s$. Clearly this proves the lemma.

This claim is established by induction on $r \geq 1$. For the inductive base let $Q = \langle C_1, \dots, C_q \rangle$ have property T . For $r = 1$, the relation \vdash satisfies $Q \vdash \emptyset$. Now suppose that $q \geq 3$. Since the C_i are distinct, at least one of C_2, C_3 must be non-empty. Without loss of generality, assume it is C_2 . Now we have a contradiction since $C_1 \vdash \emptyset \subset C_2$ and Q does not have property T . It follows that $q \leq 2$ proving the inductive base.

Assume the claim holds for all values $\leq r - 1$ and let Q have property T . We must show that $q \leq 2r^s$. Put $D = C_1$ and for each $V \subseteq D$ define the sequence Q_V as the sequence of sets $C_i - V$ such that $C_i \cap D = V$, these appearing in the same order as in Q . It is easy to show that Q_V has Property T by using methods similar to Lemma(3.3). By the inductive hypothesis, $|Q_V| \leq 2(r - 1)^{s - |V|}$ and so,

$$q = |Q| = 2 \sum_{i=0}^{|D|} \binom{|D|}{i} (r - 1)^{|D| - i}$$

i) The reader familiar with Alon and Boppana (1986) may wonder why we have: 1) Defined the Closure algorithm for sets of the form $A \cup B$ for closed A, B instead of arbitrary subsets of $U(f)$ as is done in their paper; and 2) Given a detailed exposition that S actually has property T , when this is just stated in the paper. The Closure algorithm, of course, does work for arbitrary subsets, however property T does not always hold. Consider forming the closure of $C = \{x_1 x_2\}$ when $r = 2$, $s \geq 5$ and $U(f) = P_s(\mathbf{X}_n)$. Using the Closure algorithm, $C \vdash x_1 x_2 x_j \quad \forall 3 \leq j \leq n$. Each set $x_1 x_2 x_j$ is not contained in C and is a minimal such set. It is easy to see that choosing $V_i = x_1 x_2 x_{2+i}$ is a valid choice of sequence for the closure algorithm to make. But this sequence does not have property T . $V_1, V_2 \vdash x_1 x_2 \subset V_3$. In fact in this case the number of iterations, p , is exactly $n - 2 \geq 2r^s = 2^{s+1}$ whenever $s < \lfloor \log_2 n \rfloor - 1$. Our presentation, which is sufficient for the purpose intended, avoids this problem, showing that for any such U which arises, $U \notin A \cup B$ by using the closure of A and B .

$$\leq 2 \sum_{i=0}^s \binom{s}{i} (r-1)^{s-i} = 2r^s$$

This completes the proof by induction. \square

3.5.1.3) Lower Bounds on Distance in $CLOSED(f)$

In this section we show how the combinatorial results proved in Sect(3.5.1.2) can be used to produce a general inequality for lower bounds on monotone complexity. Sect(3.5.1.4) below will then give some specific applications.

The technique used is a probabilistic counting argument, in the style of Erdős and Spencer (1974). Subsequently the following notation will be used.

$M_+(f)$ is a randomly chosen prime implicant of $f(\mathbf{X}_n)$. Each such prime implicant is selected independently with probability $|\mathbf{PI}(f)|^{-1}$.

$M_-(f)$ is a randomly chosen monom, (i.e subset of \mathbf{X}_n).

The exact details of how M_- is defined depend on the function considered.

$$Ext(f, k) = \max_{\{m \in 2^{\mathbf{X}_n} : |m|=k\}} |\{ p \in \mathbf{PI}(f) : p \leq m \}|$$

Let $A \cup B = C_0, C_1, \dots, C_p = \overline{A \cup B}$, where A and B are closed subsets of $\mathbf{U}(f)$, be the sequence of successive sets generated by the closure algorithm. Let $\langle E_1, \dots, E_p \rangle$ be the minimal sets added at each iteration. Thus $C_i \vdash E_{i+1}$ and $E_{i+1} \notin C_i$, for each $0 \leq i \leq p-1$.

$$Gap(f) = \max_{0 \leq i \leq p-1} Prob[M_-(f) \in (\lceil E_{i+1} \rceil - \lceil C_i \rceil)]$$

The reason for these random variables is to produce upper bounds on:

$$EXCESS(f) = \max_{A, B \in CLOSED(f)} Prob[M_+ \in \delta_+(A, B)] \quad (3.31)$$

$$DEFICIT(f) = \max_{A, B \in CLOSED(f)} Prob[M_- \in \delta_-(A, B)] \quad (3.32)$$

Theorem 3.15: Let $t = \rho(f, CLOSED(f))$, then

$$t \geq \frac{1 - Prob[M_+(f) \in D]}{EXCESS(f)}$$

$$t \geq \frac{Prob[M_-(f) \in D] - Prob[M_-(f) \leq f]}{DEFICIT(f)}$$

Proof: The first inequality follows from relation (3.25) in Defn(3.12), using the fact that $M_+(f)$, as a prime implicant of f , occurs in $\mathbf{I}(f)$ with probability 1. The second inequality follows from relation (3.24) in Defn(3.12). \square

$$\text{Lemma 3.19: } EXCESS(f) \leq \frac{(r-1)^{2s} Ext(f, s+1)}{|\mathbf{PI}(f)|}$$

Proof: Let $A = \lceil V \rceil$ and $B = \lceil W \rceil$ be elements of $CLOSED(f)$ where $V \subseteq \mathbf{U}(f)$, $W \subseteq \mathbf{U}(f)$ are closed. We have, from Lemma(3.16)

$$\begin{aligned} \delta_+(A, B) &= (A \cap B) - (A \sqcap B) \\ &= (\lceil V \rceil \cap \lceil W \rceil) - (\lceil V \cap W \rceil) \\ &= (\lceil base_s(V) \rceil \cap \lceil base_s(W) \rceil) - (\lceil V \cap W \rceil) \end{aligned}$$

$$\begin{aligned}
 &= \left(\bigcup_{E \in \text{base}_s(V)} \lceil E \rceil \cap \bigcup_{F \in \text{base}_s(W)} \lceil F \rceil \right) - \lceil V \cap W \rceil \\
 &= \left(\bigcup_E \bigcup_F \lceil E \rceil \cap \lceil F \rceil \right) - \lceil V \cap W \rceil \\
 &= \bigcup_E \bigcup_F \left(\lceil E \cup F \rceil - \lceil V \cap W \rceil \right)
 \end{aligned}$$

Consider any $E \in \text{base}_s(V)$ and any $F \in \text{base}_s(W)$. We can distinguish three possible cases.

Case 1: $E \cup F \not\subseteq \text{var}(p) \ \forall p \in \mathbf{PI}(f)$

In this case $\text{Prob}[M_+(f) \in \lceil E \cup F \rceil] = 0$.

Case 2: $E \cup F \in \mathbf{U}(f)$

Obviously $E \in V$, thus since V is closed $E \cup F \in V$ also. In the same way $E \cup F \in W$. So $E \cup F \in V \cap W$ and therefore the set $\lceil E \cup F \rceil - \lceil V \cap W \rceil$ is empty.

Case 3: $E \cup F \subseteq \text{var}(p)$ for some $p \in \mathbf{PI}(f)$, but $E \cup F \notin \mathbf{U}(f)$

This can only be so if $|E \cup F| \geq s+1$, by the definition of $\mathbf{U}(f)$. We now have,

$$\begin{aligned}
 \text{Prob}[M_+(f) \in \lceil E \cup F \rceil] &= \text{Prob}[E \cup F \subseteq M_+(f)] \\
 &\leq \frac{\text{Ext}(f, s+1)}{|\mathbf{PI}(f)|}
 \end{aligned}$$

So in every case, $\text{Prob}[M_+(f) \in \lceil E \cup F \rceil]$ is at most $\frac{\text{Ext}(f, s+1)}{|\mathbf{PI}(f)|}$. From the preceding analysis and Corollary(3.9),

$$\text{Prob}[M_+(f) \in \delta_+(A, B)] \leq \frac{|\text{base}_s(V)| |\text{base}_s(W)| \text{Ext}(f, s+1)}{|\mathbf{PI}(f)|}$$

$$\leq \frac{(r-1)^{2s} \text{Ext}(f, s+1)}{|\mathbf{PI}(f)|}$$

Since A, B were chosen arbitrarily the upper bound on $\text{EXCESS}(f)$ follows. \square

To produce an upper bound on $\text{DEFICIT}(f)$ we use the result of Lemma(3.18).

Lemma 3.20: $\text{DEFICIT}(f) \leq 2r^s \text{Gap}(f)$.

Proof: As in the proof of Lemma(3.19), let $A = \lceil V \rceil$, $B = \lceil W \rceil$ be elements of $\text{CLOSED}(f)$ where V, W are closed subsets of $\mathbf{U}(f)$. Let $C_0 = V \cup W$ and $\langle C_1, \dots, C_p \rangle$ be the sequence of sets created by the closure algorithm i.e the W_j sets in the description of this algorithm above. Finally let $\langle E_1, \dots, E_p \rangle$ be the sequence of minimal sets used in the closure algorithm. Recall that $C_i \vdash E_{i+1}$, $E_{i+1} \notin C_i$ and that the sequence $\langle E_1, \dots, E_p \rangle$ has property T and hence $p \leq 2r^s$. Applying Lemma(3.16) we have,

$$\begin{aligned} \delta_-(A, B) &= \lceil \overline{(V \cup W)} \rceil - \lceil V \cup W \rceil \\ &\subseteq \bigcup_{i=0}^{p-1} (\lceil C_{i+1} \rceil - \lceil C_i \rceil) \\ &= \bigcup_{i=0}^{p-1} (\lceil E_{i+1} \rceil - \lceil C_i \rceil) \end{aligned}$$

Note that the second inequality uses the fact that,

$$\bigcup_{i=0}^{k-1} (\lceil C_{i+1} \rceil - \lceil C_0 \rceil) \subseteq \bigcup_{i=0}^{k-1} (\lceil C_{i+1} \rceil - \lceil C_i \rceil)$$

This being easily established by induction on $1 \leq k \leq p$.

Now since $p \leq 2r^s$, $C_i \perp\!\!\!\perp E_{i+1}$, $E_{i+1} \notin C_i$ it is immediate from the last inequality that, the probability of $M_-(f)$ occurring in $\delta_-(A, B)$ is at most

$$2r^s \max_{0 \leq i \leq p-1} \text{Prob}[M_-(f) \in (\lceil E_{i+1} \rceil - \lceil C_i \rceil)]$$

Since A, B were arbitrary this establishes the upper bound on $DEFICIT(f)$ stated. \square

Theorem 3.16: Let $t = \rho(f, CLOSED(f))$.

$$t \geq \min \left\{ \frac{\frac{|\mathbf{PI}(f)|}{(r-1)^{2s} \text{Ext}(f, s+1)}}{\text{Prob}[M_-(f) \in D] - \text{Prob}[M_-(f) \leq f]}}{2r^s \text{Gap}(f)} \right\} \quad (3.33)$$

$$t \geq \min \left\{ \frac{\frac{|\mathbf{PI}(f)| (1 - \text{Prob}[M_+(f) \in D])}{(r-1)^{2s} \text{Ext}(f, s+1)}}{1 - \text{Prob}[M_-(f) \leq f]}}{2r^s \text{Gap}(f)} \right\} \quad (3.34)$$

Proof: (3.33) follows from Thm(3.15), Lemma(3.19) and Lemma(3.20) by considering the two cases $D = \emptyset$ and $D \neq \emptyset$. (3.34) follows in the same way by considering the two cases $D \neq \mathbf{U}(f)$ and $D = \mathbf{U}(f)$. \square

3.5.1.4) Lower Bounds for Specific Monotone Functions

We conclude this section by deriving non-trivial lower bounds on the monotone complexity of $k\text{-clique}(\mathbf{X}_n^U)$ and $PM(\mathbf{X}_{n,n})$. The first will be exponential for suitable choices of k .

We noted earlier that the actual lattice structure employed for the bound on $k\text{-clique}$ is slightly different from the family $CLOSED$ defined above. The underlying set $\mathbf{U}(f)$ is not the set of monoms

containing at most s variables, which are shortenings of prime implicants, i.e graphs with at most s edges which are subgraphs of k -cliques. Instead we take $\mathbf{U}(f)$ to be the set of all monoms corresponding to *cliques* with at most s vertices. In this way $\lceil E \rceil$, where E is a clique of size $\leq s$, is the set of graphs which contain E as a subgraph. In the same style we amend the definition of Ext , for k -cliques, to be the number of k -cliques a clique of size $s+1$ could be extended to. Since \mathbf{X}_n^U is a set of edges, each clique in $\mathbf{U}(f)$ has at least two vertices. It is not difficult to verify that the combinatorial analyses of the preceding sections all hold for the new lattice defined. For $PM(\mathbf{X}_{n,n})$ no such amendments are needed, and $CLOSED(PM)$ is exactly as defined above. We use $CLOSED(k)$ to denote the amended lattice for the k -clique function. The following is obvious and needs no proof,

Fact 3.7: i) $|\mathbf{PI}(PM(\mathbf{X}_{n,n}))| = n!$.

ii) $|\mathbf{PI}(k\text{-clique}(\mathbf{X}_n^U))| = \binom{n}{k}$. \square

To start we need upper bounds on $Ext(PM, s+1)$ and $Ext(k\text{-clique}, s+1)$

Lemma 3.21:

$$Ext(k\text{-clique}, s+1) = \binom{n-s-1}{k-s-1} \quad (3.35)$$

$$Ext(PM, s+1) = (n-s-1)! \quad (3.36)$$

Proof: (3.35) is immediate from the modified definition of $Ext(k\text{-clique}, s+1)$. For (3.36) a bipartite graph containing $s+1$ edges can only be extended to a perfect matching if each vertex has degree at most one, i.e if the graph is a perfect matching on two sets

of $s + 1$ vertices. It follows that the number of perfect matchings consistent with this is just the number of perfect matchings over two sets of $n - s - 1$ vertices. The upper bound now follows from Fact(3.7). \square

The problem of bounding $Gap(PM)$ and $Gap(k - clique)$ is more difficult. We first consider $k - clique(\mathbf{X}_n^U)$. Define $M_-(k)$ to be the following random n -vertex graph. Select a random colouring of $\{1, 2, \dots, n\}$ with g colours $\{1', 2', \dots, g'\}$, each colouring appearing independently with probability g^{-n} . For a given colouring, $\chi \in ([1..n] \rightarrow [1..g])$, $G(\chi)$ is the graph in which there is an edge between i and j if and only if $\chi(i) \neq \chi(j)$.

Lemma 3.22:

$$Gap(k - clique) \leq \left(1 - \frac{\prod_{i=0}^{s-1} (g - i)}{g^s} \right)^r$$

Proof: Given the definition of Gap , we have to show that $Prob[M_-(k) \in ([E_{i+1}] - [C_i])]$ is bounded above by the expression in the Lemma statement. Now $M_-(k)$ is a complete g -partite graph and $[E_{i+1}]$ a set of graphs containing the clique E_{i+1} as a subgraph. $M_-(k)$ contains the same clique if and only if the vertices of E_{i+1} are all coloured differently by χ , the random g -colouring which generates $M_-(k)$. Now suppose that there is some set $F \in C_i$ such that χ colours the vertices of F using different colours. In the same way $M_-(k) \in [F] \subseteq [C_i]$. A subset W of $\{1, \dots, n\}$ is said to be *properly coloured (PC)* by χ if each vertex in W is coloured differently by χ . It follows that $M_-(k) \in ([E_{i+1}] - [C_i])$ if and only if E_{i+1} is *PC* by χ but no set in C_i is *PC* by χ . So to prove the lemma it is sufficient to obtain an upper bound for,

Prob[E_{i+1} is PC by χ and no set in C_i is PC by χ]

From the definition of \perp — we can find V_1, \dots, V_r in C_i such that

$$\bigcup_{1 \leq j < l \leq r} V_j \cap V_l \subseteq E_{i+1}.$$

It follows that, *Prob[E_{i+1} is PC and no set in C_i is PC]* is no more than,

Prob[E_{i+1} is PC and V_j is not PC $\forall 1 \leq j \leq r$]

which does not exceed

$$\text{Prob}[V_j \text{ is not PC} \mid E_{i+1} \text{ is PC}]$$

and this is at most

$$\prod_{j=1}^r \text{Prob}[V_j \text{ is not PC} \mid E_{i+1} \text{ is PC}]$$

The last inequality holds by virtue of the fact that the sets $V_j - E_{i+1}$ are disjoint (by definition of \perp —) and hence the events $\langle V_j \text{ is not PC} \mid E_{i+1} \text{ is PC} \rangle$ are mutually independent. Now let $p_j = |V_j \cap E_{i+1}|$ and $q_j = |V_j - E_{i+1}|$ so that $p_j + q_j = |E_{i+1}| \leq s$.

$$\text{Prob}[V_j \text{ is not PC} \mid E_{i+1} \text{ is PC}] = 1 - \text{Prob}[V_j \text{ is PC} \mid E_{i+1} \text{ is PC}]$$

$$= 1 - \frac{\prod_{l=p_j}^{|E_{i+1}|-1} (g-l)}{g^{q_j}}$$

$$\leq 1 - \frac{\prod_{l=p_j}^{s-1} (g-l)}{g^{s-p_j}}$$

$$\leq 1 - \frac{\prod_{l=0}^s (g-l)}{g^s}$$

This proves the lemma. \square

Lemma 3.23: Let $g = k - 1$. If $D \in \text{CLOSED}(k)$ and $D \neq \emptyset$ then

$$\text{Prob}[M_-(k) \leq k - \text{clique}] = 0$$

$$\text{Prob}[M_-(k) \in D] \geq \frac{\prod_{i=0}^{s-1} (k-1-i)}{(k-1)^s}$$

Proof: The first inequality is obvious. For the second since $D \neq \emptyset$, D contains at least one set E , say. Thus

$$\text{Prob}[M_-(k) \in D] \geq \text{Prob}[M_-(k) \in [E]]$$

$$= \text{Prob}[E \subseteq M_-(k)]$$

$$= \text{Prob}[E \text{ is PC by } \chi \text{ s.t. } G(\chi) = M_-(k)]$$

and this proves the second inequality. \square

Theorem 3.17: For $3 \leq k \leq \frac{1}{4} \left(\frac{n}{\log n} \right)^{2/3}$,

$$\mathbf{C}^{\mathbf{m}}(k - \text{clique}) = \Omega \left(\left(\frac{n}{16k^{3/2} \log n} \right)^{\sqrt{k}} \right)$$

Proof: Fix $s = \lceil \sqrt{k} \rceil$, $r = \lceil 4 \sqrt{s \log n} \rceil + 1$, $g = k - 1$. This gives from Thm(3.16) (2.11), using $t = \rho(\text{CLOSED}(k), k - \text{clique})$,

$$\begin{aligned} t &\geq \frac{\binom{n}{k}}{(r-1)^{2s} \binom{n-s-1}{k-s-1}} \\ &= \frac{n! (k-s-1)!}{(n-s-1)! k! (r-1)^{2s}} \\ &\geq \frac{n^n k^{k-s}}{n^{n-s} k^k (r-1)^{2s}} \\ &\geq \left(\frac{n}{k (r-1)^2} \right)^s \\ &\geq \left(\frac{n}{16k^{3/2} \log n} \right)^{\sqrt{k}} \end{aligned}$$

We leave as an exercise the problem of showing that t exceeds this quantity in the case $D \neq \emptyset$ in (2.11) of Thm(3.16). \square

For the function $PM(\mathbf{X}_{n,n})$, the random monom (\equiv bipartite graph) M_- is constructed by the method below.

Let V, W be the disjoint sets of n vertices in the bipartite graph $B(\mathbf{X}_{n,n})$. Select a random labelling, h , of the vertices $V \cup W$ with 0 and 1. Each labelling is chosen with probability 2^{-2n} . $M_-(PM)$ is the random bipartite graph formed by choosing such a labelling of $V \cup W$ and adding edges $\{ \langle v_i, w_j \rangle : h(v_i) = h(w_j) \}$, where $1 \leq i, j \leq n$.

In order to bound $Gap(PM)$ with this choice of M_- , Razborov (1985b) proves some combinatorial results on properties of $\mathbf{U}(PM)$. It should be noted that in graph-theoretic terms, an element of $\mathbf{U}(PM)$ corresponds to a matching containing at most s edges. A *matching* is a (bipartite) graph in which every vertex is the endpoint of at most one edge. This interpretation is convenient for developing an upper bound on $Gap(PM)$.

Lemma 3.24: (Razborov, 1985b) Let

$$\mathbf{B} = \{B_1, B_2, \dots, B_r\} \subseteq \mathbf{U}(PM)$$

be a set of r non-empty matchings such that $B_i \cap B_j = \emptyset$ whenever $i \neq j$. There is a subset

$$\{T_1, T_2, \dots, T_p\}$$

of \mathbf{B} such that $p \geq \frac{\sqrt{r}}{s}$ and for which the bipartite graph with edges $\bigcup_{i=1}^p T_i$ contains no cycles, i.e. is a *forest*.

Proof: Let $\{T_1, T_2, \dots, T_p\}$ be a maximal size subset of \mathbf{B} for which $\bigcup_{i=1}^p T_i$ is a forest. It suffices to prove that $p \geq \frac{\sqrt{r}}{s}$. Suppose the contrary and put $E_0 = \bigcup_{i=1}^p T_i$. Since $|T_i| \leq s$, by the assumption we have $|E_0| < \sqrt{r}$. Now consider the subsets V_0 of V , W_0 of W , being those vertices in V , respectively W , which occur in at least one edge of E_0 .

Clearly $|V_0| < \sqrt{r}$ and $|W_0| < \sqrt{r}$ hence $|V_0 \times W_0| < r = |\mathbf{B}|$. It follows from the edge disjointness of matching in \mathbf{B} that we can find some matching $B_j \in \mathbf{B}$ such that $B_j \cap (V_0 \times W_0) = \emptyset$. By definition $E_0 \subseteq V_0 \times W_0$ hence $B_j \cap E_0 = \emptyset$. But E_0 is a forest and B_j a matching and so from the preceding argument the graph with edges $E_0 \cup B_j$ is also a forest. This contradicts the choice of $\{T_1, \dots, T_p\}$ as being maximal and therefore we must have $p \geq \frac{\sqrt{r}}{s}$. \square

Lemma 3.25: Let T be a forest over $V \cup W$ which contains exactly p edges $\{ \langle i_k, j_k \rangle : 1 \leq k \leq p \}$. The events $\{ \langle i_k, j_k \rangle \text{ is an edge of } M_-(PM) \}$ (for each k) occur independently with probability $1/2$.

Proof: It is sufficient to show that for any subset K of $\{1, 2, \dots, p\}$ the probability of the event

$$\forall k \in K \langle i_k, j_k \rangle \in M_-(PM) ; \forall k \notin K \langle i_k, j_k \rangle \notin M_-(PM)$$

is exactly 2^{-p} .

Let $\chi_K : \{1, \dots, n\} \rightarrow \{0, 1\}$ be the predicate for which $\chi_K(k) = 1 \iff k \in K$. Now recalling that $M_-(PM)$ arises from a random labelling $h : V \cup W \rightarrow \{0, 1\}$ it is clear that the probability of this event is just the number of labellings, h , which are solutions to the following system of p linear equations over $\mathbf{GF}(2)$, divided by 2^{-2n} , i.e the total number of distinct labellings.

$$\left\{ h(v_{i_k}) \oplus h(w_{j_k}) \equiv \chi_K(k) \oplus 1 \right\}_{1 \leq k \leq p}$$

So it suffices to show that this system has 2^{2n-p} distinct solutions.

Consider the forest T with p edges $\{ \langle i_k, j_k \rangle : 1 \leq k \leq p \}$. Let $\beta \geq 1$ be the number of connected components (i.e trees) in this T , each component containing at least one edge. Then T contains exactly $p + \beta$ vertices. For each tree there are exactly 2 ways of labelling the vertices to satisfy the system of equations above, one being the logical complement of the other. That there are exactly two such consistent labelling can be proved by an easy induction on the number of edges in a single component. Now since each component may be labelled independently of the others it follows that there are 2^β labelling of the vertices in the forest T which satisfy the system. This leaves $2n - p - \beta$ vertices unlabelled (those not the endpoint of any edge in T) and any labelling of these will be valid. Thus the system of linear equations over $\mathbf{GF}(2)$ has exactly $2^\beta \cdot 2^{2n - p - \beta} = 2^{2n - p}$ distinct solutions as required. \square

Corollary 3.10: Let $D \in \text{CLOSED}(PM)$ with $D \neq \emptyset$. $\text{Prob}[M_- \in \lceil D \rceil] \geq 2^{-s}$.

Proof: Since D is non-empty it contains at least one matching, E say. Note that E is obviously a forest. We therefore have

$$\begin{aligned} \text{Prob}[M_- \in \lceil D \rceil] &\geq \text{Prob}[M_- \lceil E \rceil] \\ &= \text{Prob}[E \subseteq M_-] = 2^{|E|} \geq 2^{-s} \end{aligned}$$

\square

Lemma 3.26: $\text{Gap}(PM) \leq (1 - 2^{-s})^{\frac{\sqrt{r}}{s}}$

Proof: From the definition of $\text{Gap}(f)$ it is sufficient to show that if $C \subseteq \mathbf{U}(PM)$ and $C \dashv\vdash E$ then

$$Prob[M_-(PM) \in (\lceil E \rceil - \lceil C \rceil)] \leq (1 - 2^{-s})^{\frac{\sqrt{r}}{s}}$$

So suppose that $C \subseteq U(PM)$ and we have

$$\langle E_1, \dots, E_r \rangle \in \{C\}^r$$

for which

$$\langle E_1, \dots, E_r \rangle \perp\!\!\!\perp E$$

Consider the set (of matchings) $\{F_i : F_i = E_i - E\}$. From (3.30), the definition of $\perp\!\!\!\perp$, we have $F_i \cap F_j = \emptyset$ whenever $i \neq j$. In addition if any F_i is empty then $E_0 \subseteq E_i$ and hence $\lceil E_0 \rceil \subseteq \lceil C \rceil$ for which the upper bound on $Gap(PM)$ claimed follows trivially. So it may be assumed that each F_i ($1 \leq i \leq r$) is non-empty. Now the conditions of Lemma(3.24) holds for the set $\{F_1, \dots, F_r\}$ thus we can find a subset $T = \{T_1, \dots, T_p\}$ of this such that $p \geq \frac{\sqrt{r}}{s}$ and for which $\bigcup_{j=1}^p T_j$ is a forest.

We now have,

$$Prob[M_- \in (\lceil E \rceil - \lceil C \rceil)]$$

is no more than

$$Prob[M_- \in \lceil E \rceil \ \& \ \forall i \ M_- \notin \lceil E_i \rceil]$$

and this is equal to

$$Prob[E \subseteq M_- \ \& \ \forall i \ F_i \subseteq M_-] \leq Prob[\forall i \ F_i \subseteq M_-]$$

$$\leq Prob[\forall T_j \in T \ T_j \subseteq M_-]$$

From Lemma(3.25) the events $T_j \subseteq M_-$ (for each $1 \leq j \leq p$) are independent and occur with probability $2^{T_j} \geq 2^{-s}$. Therefore

$Prob[\forall T_j \in T T_j \subseteq M_-]$ is equal to $\prod_{j=1}^p Prob[T_j \subseteq M_-]$ and hence

is no more than $(1 - 2^{-s})^{\frac{\sqrt{r}}{s}}$. This establishes the upper bound on $Gap(PM)$. \square

Lemma 3.27: $Prob[M_- \leq PM(\mathbf{X}_{n,n})] \leq \frac{1}{\sqrt{n}}$.

Proof: M_- contains a perfect matching if and only if the number of vertices of V labelled with 1 by a random labelling h equals the number of vertices of W labelled 1 by the same random labelling. Thus,

$$\begin{aligned} Prob[M_- \leq PM(\mathbf{X}_{n,n})] &\leq Prob[\sum_{i=1}^n h(v_i) = \sum_{i=1}^n h(w_i)] \\ &\leq \max_{0 \leq j \leq n} Prob[\sum_{i=1}^n h(w_i) = j] \\ &= \binom{n}{n/2} 2^{-2n} \\ &\leq \frac{1}{\sqrt{n}} \end{aligned}$$

\square

Theorem 3.18: For any $\varepsilon > 0$ and n sufficiently large,

$$C^m(PM(\mathbf{X}_{n,n})) \geq n^{(\frac{1}{16} - \varepsilon) \log n}$$

Proof: Fix $s = \lfloor \log n/8 \rfloor$ and $r = \lfloor n^{1/4}(\log n)^8 \rfloor$ and let $t = \rho(CLOSED(PM), PM)$. Using relation (3.33) of Thm(3.16) and relation (3.36) of Lemma(3.21) gives,

$$\begin{aligned}
t &\geq \frac{n!}{(r-1)^{2s} (n-s-1)!} \\
&\geq \left[\frac{n}{(r-1)^2} \right]^s \\
&\geq n^{\left(\frac{1}{16} - \varepsilon\right) \log n}
\end{aligned}$$

Now consider the second part of relation (3.33) in Thm(3.16), i.e. D is non-empty. Using Corollary(3.10), Lemma(3.25), Lemma(3.26) and Lemma(3.27) and the chosen values of r and s shows that in this case t would be at least $\exp(\log^3 n - o(\log^3 n))$ and hence t is asymptotically greater than the first case, $D = \emptyset$. This proves the theorem. \square

An important consequence of Thm(3.18) concerns the power of negation in computing Boolean functions.

Corollary 3.11: The basis $\{\wedge, \vee, \neg\}$ is superpolynomially more powerful than the basis $\{\wedge, \vee\}$.

Proof: $PM(\mathbf{X}_{n,n})$ can be computed using polynomial size networks over any logically complete basis e.g by combining the algorithm of Hopcroft and Karp (1973) with the result of Corollary(2.1) above. Thm(3.18) shows that polynomial size monotone networks do not exist for this function. \square

Tardos (1988) has recently shown that the gap between monotone and non-monotone network complexity is in fact exponential.

3.5.2. The Andreev Lower Bound Method

The techniques applied in Andreev (1985) are developed from the classical inductive gate elimination method and utilise Wegener’s idea of providing certain functions "for free" as additional inputs. This approach is shown to yield an exponential bound of $2^{n^{1/8} - o(1)}$ on the monotone complexity of a specific function in M_n .

Below E^n denotes the set $\{0, 1\}^n$, $\|\alpha\|$ the number of 1’s in $\alpha \in E^n$ and D_f the (minimal) DNF of $f \in M_n$. The size of f (denoted by $|f|$) is the number of prime implicants of f ; The rank of f (Rf) is the length of the longest prime implicant of f . If f is a constant function then $|f| = Rf = 0$. Given f_1 and f_2 in M_n , we say that $f_1 \subseteq f_2$ if and only if $\mathbf{PI}(f_1) \subseteq \mathbf{PI}(f_2)$. A function, f , is called (u, r) -regular if it can be expressed in the form:

$$f = x_{i_1} \wedge x_{i_2} \wedge \cdots \wedge x_{i_u} \wedge f_1$$

Here i_1, \dots, i_u are distinct, f_1 does not depend on $\{x_{i_1}, \dots, x_{i_u}\}$, $|f_1| = r$ and each dependent variable of f_1 occurs in D_{f_1} exactly once. f is called r -regular if it is (u, r) -regular for some $u \geq 0$.

M_n^t denotes those functions $f \in M_n$ such that every prime implicant of f has length t . Define:

$$\pi_t(f) = \begin{cases} \min_{g \in M_n^t : f \leq g} |g| & \text{if such a } g \text{ exists} \\ 1 + \binom{n}{t} & \text{otherwise} \end{cases}$$

$$l(r, s) = r^s s!$$

$$R_{r,s}^n = \{ f : f \in M_n, Rf \leq s, |f| \leq l(r, s) \}$$

An (n, r, s) -scheme is a monotone network with functions from $R_{r,s}^n$

given free as extra inputs. $L_{r,s}^n$ is the least number of \wedge, \vee gates needed to realise $f \in M_n$ by a (n, r, s) -scheme. It may be assumed that $r \geq 2$ and $s \geq 1$. Clearly, $L_{r,s}^n(f) = 0 \iff f \in R_{r,s}^n$.

Let $0 < p < 1$. For f_1 and f_2 in M_n we define a measure $\rho_p(f_1, f_2)$ as follows:

$$\rho_p(f_1, f_2) = \sum_{\alpha \in E^n : f_1(\alpha) \neq f_2(\alpha)} p^{n - \|\alpha\|} (1 - p)^{\|\alpha\|}$$

$\rho_p(f, g)$ may be interpreted in the following way. Consider constructing a random member, β , of E^n by setting x_i to 0 with probability p and to 1 with probability $1 - p$, the events $\{x_i = e : e \in \{0, 1\}\}$ for $1 \leq i \leq n$ being independent. In this way $\rho_p(f, g)$ is just the probability that $f(\beta) \neq g(\beta)$. It is easy to see that,

$$\rho_p(F(\mathbf{X}_n, f_1), F(\mathbf{X}_n, f_2)) \leq \rho_p(f_1, f_2) \quad (3.37)$$

Lemma 3.28: Let $\{i_1, \dots, i_u\} \subseteq \{1, \dots, n\}$ and suppose that

$$g(\mathbf{X}_n) = \bigwedge_{j=1}^u x_{i_j} \wedge g_1(\mathbf{X}_n - \{x_{i_j} : 1 \leq j \leq u\})$$

where g_1 is $(0, r)$ -regular.

If $Rg \leq s$ then $\rho_p(g, x_{i_1} \cdots x_{i_u}) \leq (sp)^r$.

Proof: Without loss of generality suppose that $g = x_1 x_2 \cdots x_u \wedge g_1$, where g_1 is $(0, r)$ -regular. By definition every variable on which g_1 essentially depends occurs in D_{g_1} exactly once and $|g_1| = r$. Additionally since $Rg \leq s$ it is obvious that $Rg_1 \leq s$ also. Hence

$$\begin{aligned} \rho_p(1, g_1) &= \prod_{m \in \mathbf{PI}(g_1)} \rho_p(1, m) \\ &= \prod_{m \in \mathbf{PI}(g_1)} \sum_{x \in \text{var}(m)} p \\ &\leq (sp)^r \end{aligned}$$

Thus from (3.37) $\rho_p(g, x_1 x_2 \cdots x_u) \leq (sp)^r$, by using $F = f_i \wedge x_1 \wedge \cdots \wedge x_u$. \square

Lemma 3.29: If $f \in M_n$ and $Rf \leq s$, then there exists some function \hat{f} in $R_{r,s}^n$ such that $f \leq \hat{f}$ and $\rho_p(\hat{f}, f) \leq |f| (sp)^r$.

Proof: First of all suppose that if $f \in M_n$, $Rf \leq s$ and $|f| \geq l(r, s)$ then there exists an r -regular g such that $g \subseteq f$. Using the result of Lemma(3.28), we can construct a sequence of functions $f \equiv f_0, f_1, \dots, f_t \equiv \hat{f}$ which for $0 \leq i < t$ satisfy:

- i) $|f_i| \geq l(r, s)$
- ii) $\rho_p(f_i, f_{i+1}) \leq (sp)^r$

This sequence can be constructed by the following procedure.

```

i := 0 ; f0 := f
while |fi| ≥ l(r, s) do begin
    Find an r-regular g such that g ⊆ fi,
    {thus g ≡ m ∧ g1, say}
    i := i + 1
    fi := fi-1 ∨ m
od
    
```

Note that this procedure terminates because $|f_{i+1}| < |f_i|$. Obviously (i) is satisfied. Also each f_i is of the form $f_i = h_i \vee m_i \wedge g_i$ where $m_i \wedge g_i$ is r -regular. In this way $f_{i+1} = h_i \vee m_i$. Thus,

$$\begin{aligned}
\rho_p(f_i, f_{i+1}) &= \rho_p(h_i \vee m_i \wedge g_i, h_i \vee m_i) \\
&\leq \rho_p(m_i \wedge g_i, m_i) \quad (\text{by 3.1}) \\
&\leq (sp)^r \quad \text{by Lemma(3.28)}
\end{aligned}$$

Since,

$$\rho_p(f, f_i) \leq \rho_p(f, f_{i-1}) + \rho_p(f_{i-1}, f_i)$$

The final function obtained is the \hat{f} of the Lemma statement.

So it suffices to prove that the supposition stated at the start of the proof does in fact hold. We prove this by induction on s .

If $s = 1$ then f itself is $|f|$ -regular i.e trivially g exists. Now suppose the result holds for $s \leq t - 1$. Let $s = t$ and f_1 be a $(0, |f_1|)$ -regular function and the maximal possible such that $f_1 \subseteq f$. If $|f_1| \geq r$ then trivially g exists. Therefore suppose that $|f_1| \leq r - 1$. Without loss of generality let x_1, \dots, x_k be the variables which f_1 depends upon. It is clear that $k \leq s$. By the maximality of f_1 every prime implicant of f contains at least one of x_1, \dots, x_k . Without loss of generality suppose that x_1 occurs in the largest number of prime implicants, and let f_2 be the disjunction over all these prime implicants. Then $f_2 = x_1 \wedge f_3$ where f_3 does not depend on x_1 . Clearly $Rf_3 = Rf_2 - 1 \leq t - 1$. Also

$$|f_3| = |f_2| \geq |f|/k \geq r^{t-1} (t-1)! = l(r, t-1)$$

So by the inductive hypothesis, there exists an r -regular g_3 such that $g_3 \subseteq f_3$. So if $g = x_1 \wedge g_3$ then g is r -regular and $g \subseteq f_2 \subseteq f$. \square

Lemma 3.30: If $f \in M_n$ such that $L_{r,s}^n(f) > 0$ then there exists some $g \in M_n$ such that:

- i) $\rho_p(1, g) \geq \rho_p(1, f) - (sp)^r l(r, s)^2$
- ii) $\pi_{s+1}(g) \geq \pi_{s+1}(f) - l(r, s)^2$
- iii) $L_{r,s}^n(g) \leq L_{r,s}^n(f) - 1$

Proof: Let S be an optimal (n, r, s) -scheme realising f . Consider any gate both of whose inputs are inputs h_1, h_2 in $R_{r,s}^n$ of S . Both are non-constant by the assumption of optimality. The output of this gate is some function $h_1 * h_2$, where $*$ = \wedge or \vee . It is clear that in either case

$$|h_1 * h_2| \leq \max \{ |h_1|, |h_2|, |h_1| + |h_2| \} \leq l(r, s)^2 \quad (3.38)$$

Let h_3 and h_4 be functions whose minimal DNF contains each prime implicant of $D_{h_1 * h_2}$ of length more than s , respectively not more than s . Let h be an arbitrary function in M_n . Consider a network $S(h)$ which is obtained from S by removing the output of this gate and replacing it with the function h . Let $G(h)$ denote the function computed by $S(h)$. It is easy to see that:

$$G(h_4) \leq f \leq G(h_4) \vee h_3 \quad (3.39)$$

Therefore: $\rho_p(1, G(h_4)) \geq \rho_p(1, f)$. From Lemma(3.29) we can find a function $h_5 \in R_{r,s}^n$ such that

$$h_4 \leq h_5 \quad ; \quad \rho_p(h_4, h_5) \leq (sp)^r |h_4| \quad (3.40)$$

Set $g = G(h_5)$, from (3.37) and (3.38) it follows that:

$$\rho_p(g, G(h_4)) \leq (sp)^r l(r, s)^2$$

Using the triangle inequality we have:

$$\rho_p(1, g) \geq \rho_p(1, G(h_4)) - \rho_p(g, G(h_4)) \quad (3.41)$$

$$\geq \rho_p(1, f) - (sp)^r l(r, s)^2$$

Now from (3.39) and (3.40),

$$f \leq g \vee h_3 \quad ; \quad \pi_{s+1}(f) \leq \pi_{s+1}(g) + \pi_{s+1}(h_3)$$

Applying (3.38) and the fact that each prime implicant in D_{h_3} contains at least $s + 1$ variables it follows that $\pi_{s+1}(h_3) \leq |h_3| \leq l(r, s)^2$ and so

$$\pi_{s+1}(g) \geq \pi_{s+1}(f) - \pi_{s+1}(h_3) \geq \pi_{s+1}(f) - l(r, s)^2 \quad (3.42)$$

Clearly,

$$L_{r,s}^n(g) \leq L_{r,s}^n(f) - 1 \quad (3.43)$$

(3.41), (3.42) and (3.43) prove the result. \square

Theorem 3.19: If $f \in M_n$ then

$$L_{r,s}^n(f) \geq \frac{1}{l(r, s)^2} \min \left\{ \pi_{s+1}(f), \frac{\rho_p(1, f) - sp}{(sp)^r} \right\}$$

Proof: If $f \in R_{r,s}^n$ then the assertion holds since both sides of this inequality are ≤ 0 . If $L_{r,s}^n(f) > 0$ then from Lemma(3.30) there exists a sequence of functions g_1, g_2, \dots, g_t in M_n such that:

$$\rho_p(1, g_i) \geq \rho_p(1, f) - il(r, s)^2 (sp)^r$$

$$\pi_{s+1}(g_i) \geq \pi_{s+1}(f) - il(r, s)^2 \quad i = 1, 2, \dots, t$$

$$L_{r,s}^n(f) > L_{r,s}^n(g_1) > \dots > L_{r,s}^n(g_t) = 0$$

Clearly g_t is a member of $R_{r,s}^n$. If $g_t \equiv 0$ then $\pi_{s+1}(g_t) = 0$ and consequently,

$$t = \frac{\pi_{s+1}(f)}{l(r, s)^2} \tag{3.44}$$

If $g_t \neq 0$ then it follows that $Rg_t \leq s$ and then $\rho_p(1, g_t) \leq sp$. Thus $\rho_p(1, f) - tl(r, s)^2 (sp)^r \leq sp$. So,

$$t \geq \frac{\rho_p(1, f) - sp}{l(r, s)^2 (sp)^r} \tag{3.45}$$

(3.44) and (3.45) give the result. \square

Let $T = [m_{ij}]$, where $1 \leq i \leq m$, $1 \leq j \leq n$, be an $m \times n$ Boolean matrix without any zero rows. Define the function f_T by,

$$f_T(x_1, \dots, x_n) = \bigvee_{i=1}^m \bigwedge_{j: m_{ij}=1} x_j$$

Corollary 3.12: If every row of the Boolean (m, n) -matrix T contains at least $t \geq s + 1$ 1's and T does not have a $(k, s + 1)$ -submatrix of 1's then,

$$L_{r,s}^n(f_T) \geq \frac{1}{l(r, s)^2} \min \left\{ \frac{m}{k-1}, \frac{1 - me^{-pt} - sp}{(sp)^r} \right\}$$

This follows since:

$$\pi_{s+1}(f_T) \geq \frac{m}{k-1}$$

and

$$\rho_p(1, f_T) \geq 1 - m(1-p)^s \geq 1 - me^{-pt} \quad \square$$

Let $GF(q)$ be the Galois field of order q and let $n = q^2$. Also let the pairs (α_j, β_j) range over the set $GF(q) \times GF(q)$. F_i , where $i = 1, 2, \dots, m = q^{s+1}$, is an enumeration of all polynomials over $GF(q)$ whose degree does not exceed s . $T_{n,s}$ denotes the

(m, n) -Boolean matrix such that

$$m_{ij} = 1 \iff \alpha_j = F_i(\beta_j)$$

It is easy to see that for $i_1 \neq i_2$ the system of equations:

$$y = F_{i_1}(x) ; y = F_{i_2}(x)$$

have no more than s common solutions; Consequently the matrix $T_{n,s}$ does not have a $(2, s+1)$ -submatrix of 1's. In addition each row of $T_{n,s}$ contains exactly q 1's. Set $f_{n,s} = F_{T_{n,s}}$ and from Corollary(3.1) we have

Corollary 3.13:

$$L_{r,s}^n(f_{n,s}) \geq \frac{1}{l(r,s)^2} \min \left\{ n^{(s+1)/2}, \frac{1 - n^{(s+1)/2} e^{-p\sqrt{n}} - sp}{(sp)^r} \right\} \quad \square$$

If we fix

$$s \leq \frac{0.5n^{1/8}}{\log_e n - 2}$$

$$p = \frac{(s+1) \log_e n + 2}{2\sqrt{n}}$$

$$r = \lfloor (s+1) \log_e n \rfloor$$

This gives

Corollary 3.14:

$$L_{r,s}^n(f) \geq \left(\frac{\sqrt{n}}{4s^4 \log_e^2 n} \right)^{s+1} \quad \square$$

This evaluates to

$$\exp\left(\frac{n^{1/8} \log_e \log_e n}{\log_e n}\right)$$

3.5.3. Conclusion

Andreev (1985) and the work of Razborov (1985a,b) and (Alon and Boppana, 1986) offer two approaches to proving non-trivial lower bounds on monotone network size. In this final section we consider in what ways the two basic methods are similar.

Both techniques are, in a broad sense, inductive arguments based on Wegener’s concept of providing functions for free as additional inputs. However this is only explicit in Andreev’s proof. In the Lattice method the inductive argument occurs in the general lemma relating monotone network size to the distance metric in regular lattices, i.e Lemma(3.1), but is not otherwise applied in the subsequent combinatorial analyses. The elements of the lattice $CLOSED(f)$ (and by extension $CLOSED(k)$ for k -clique) correspond to particular *monotone* Boolean functions. Thus we can define a mapping $REPR : CLOSED(f) \rightarrow M_n$ as follows; For $A \in CLOSED(f)$, $\mathbf{I}(REPR(A)) = A$ so that

$$\mathbf{PI}(REPR(A)) = base_s(A)$$

Note that the cover operation $\lceil \dots \rceil$ ensures that $REPR(A)$ is in fact a monotone function. The set of functions $\{REPR(A) : A \in CLOSED(f)\}$ form a subset of the set $R_{r,s}^n$ used by Andreev. This is a proper subset since $|\mathbf{PI}(REPR(A))| \leq (r-1)^s$ while for $f \in R_{r,s}^n$ we have only $|\mathbf{PI}(f)| \leq l(r,s) = r^s s!$.

At first sight it appears that the novel closure relation used by Razborov does not seem to have any analogue in Andreev’s proof. In fact Razborov (pers. comm.) has pointed out that this is not the case.

Recall Lemma(3.29) in the derivation of Andreev's proof:

If $f \in M_n$ and $Rf \leq s$ then there is a function, $\hat{f} \in R_{r,s}^n$ such that $\hat{f} \leq f$ and $\rho_p(\hat{f}, f) \leq |f|(sp)^r$.

Consider the sets of monoms $U_0 = \mathbf{PI}(f)$ and $\hat{U}_0 = \mathbf{PI}(\hat{f})$ so that

$$\lceil U_0 \rceil = \mathbf{I}(f) ; \lceil \hat{U}_0 \rceil = \mathbf{I}(\hat{f})$$

If we examine the proof of this lemma and the process by which \hat{f} is constructed from f it turns out that \hat{U}_0 consists of the minimal sets in $\overline{U_0}$, i.e the closure of U_0 . So clearly

$$\mathbf{I}(\hat{f}) = \lceil \hat{U}_0 \rceil = \lceil \overline{U_0} \rceil$$

and thus from the earlier bound on the number of minimal sets in a closed set, proved in Razborov (1985b),

$$|\mathbf{PI}(\hat{f})| \leq r^s s! \quad \Leftrightarrow \quad \hat{f} \in R_{r,s}^n$$

In addition, $U_0 \subseteq \overline{U_0}$, hence

$$\begin{aligned} \mathbf{I}(f) &= \lceil U_0 \rceil \subseteq \lceil \overline{U_0} \rceil \\ &= \lceil \hat{U}_0 \rceil = \mathbf{I}(\hat{f}) \end{aligned}$$

showing that $f \leq \hat{f}$. Finally we have the relation

$$\rho_p(\hat{f}, f) \leq |f|(sp)^r$$

which, from the previous development, reduces to

$$\rho_p(\hat{f}, f) = \text{Prob} [M_- \in \lceil \overline{U_0} \rceil - \lceil U_0 \rceil]$$

where M_- is a random monom defined by the distribution

$$\text{Prob} [M_- \leq x] = 1 - p \text{ independently for all } x \in \mathbf{X}_n$$

It has already been shown that the RHS of this equality is bounded above by the number of iterations of the closure algorithm multiplied by $Gap(f)$. Andreev applies the simple upper bound that the number of iterations does not exceed $|f|$ and $Gap(f)$, with the choice of M_- is easily shown to be $\leq (sp)^r$.

In summary the closure operation employed by Razborov and its properties are paralleled in the proof and statement of Lemma(3.29) used by Andreev.

In fact it turns out that we may improve Andreev's lower bound inequality by recasting his proof in terms of the functions arising from *REPR* instead of the set $R_{r,s}^n$ and by appealing to the improved combinatorial analyses of Alon and Boppana (1986). Below we describe how this is accomplished.

We shall call a monotone Boolean network, S , an (n,r,s,d) -scheme, where $d \in M_n$, if S has as inputs exactly the set of monotone functions

$$R_{r,s}^{n,d} = \{ REPR(\lceil A \rceil) : \lceil A \rceil \in CLOSED(d) \}$$

Since $CLOSED(d)$ is a regular lattice this provides the normal network inputs \mathbf{X}_n . $L_{r,s}^{n,d}(f)$ denotes the number of gates in a minimal (n,r,s,d) -scheme realising f . It should be noted that d is not required to equal f . The quantities $\pi_i(f)$ and ρ_p retain their meanings of Section(3.5.2). The main result to be reproved is Lemma(3.30), which can be sharpened by using Lemma(3.16), Corollary(3.9), Lemma(3.18) and Lemma(3.20). We can dispense entirely with Lemma(3.28) and Lemma(3.29) of Andreev (1985) in the course of this proof.

Lemma 3.31: If $f \in M_n$ such that $L_{r,s}^{n,d}(f) > 0$ then there exists some $g \in M_n$ for which,

- i) $\rho_p(1, g) \geq \rho_p(1, f) - 2r^s (sp)^r$
- ii) $\pi_{s+1}(g) \geq \pi_{s+1}(f) - (r-1)^{2s}$
- iii) $L_{r,s}^{n,d}(g) \leq L_{r,s}^{n,d}(f) - 1$

Proof: The proof parallels that of Lemma(3.30). Let S be an optimal (n, r, s, d) -scheme realising f . Consider any gate of S whose inputs are functions $h_1, h_2 \in R_{r,s}^{n,d}$. By definition

$$h_1 = REPR(\lceil A \rceil) \quad ; \quad h_2 = REPR(\lceil B \rceil)$$

for some closed subsets, A and B , of $\mathbf{U}(d)$. The output of the selected gate is some function $h_1 * h_2$ where $*$ = \wedge or \vee . Thus, from Corollary(3.9).

$$|h_1 * h_2| \leq \max \{|h_1|, |h_2|, |h_1| + |h_2|\} \leq (r-1)^{2s} \quad (3.46)$$

Let h_3 , resp. h_4 , be functions whose minimal DNF consists of all prime implicants of $h_1 * h_2$ having length more than, resp. at most, s variables. For any $h \in M_n$, $S(h)$ denotes the (n, r, s, d) -scheme obtained by replacing the gate computing $h_1 * h_2$ in S , by a node computing h . $G(h)$ denotes the function computed by $S(h)$. It is obvious that

$$G(h_4) \leq f \leq G(h_4) \vee h_3 \quad (3.47)$$

and so $\rho_p(1, G(h_4)) \geq \rho_p(1, f)$.

At this point the proof diverges from Lemma(3.30).

We claim there is some function $h_5 = REPR(\lceil C \rceil) \in R_{r,s}^{n,d}$ for which,

$$h_4 \leq h_5 \quad ; \quad \rho_p(h_4, h_5) \leq \begin{cases} 0 & \text{if } * = \wedge \\ 2r^s (sp)^r & \text{if } * = \vee \end{cases} \quad (3.48)$$

First consider $\ast = \wedge$. In this case choosing h_5 to be $REPR(\lceil A \rceil \sqcap \lceil B \rceil)$ satisfies (3.48). To show this it is sufficient to prove that $h_4 = h_5$. Now, using Lemma(3.16),

$$\mathbf{PI}(h_4) = \mathbf{PI}(REPR(\lceil A \rceil) \wedge REPR(\lceil B \rceil)) \cap P_s(\mathbf{X}_n)$$

$$\mathbf{PI}(h_5) = base_s(\lceil A \rceil \sqcap \lceil B \rceil) = base_s(\lceil A \cap B \rceil)$$

Suppose $m \in \mathbf{PI}(h_4)$. Then $|var(m)| \leq s$ and there are monoms

$$m_1 \in base_s(\lceil A \rceil); m_2 \in base_s(\lceil B \rceil)$$

for which $m = m_1 \wedge m_2$. Since, $|var(m)| \leq s$, $var(m_i) \subseteq var(m)$ and A, B are closed subsets of $\mathbf{U}(d)$ it follows that $m \in A, m \in B$ thus $m \in \lceil A \cap B \rceil$ and so $m \leq h_5$. On the other hand suppose that $m \in \mathbf{PI}(h_5)$. Then $m \in \lceil A \cap B \rceil$ and $|var(m)| \leq s$. This implies the existence of some $m_3 \in A \cap B$ for which $var(m_3) \subseteq var(m)$ and hence $m \in A, m \in B$ by closure. It follows that $m \leq REPR(\lceil A \rceil) \wedge REPR(\lceil B \rceil)$ and since $|var(m)| \leq s$ we have therefore $m \leq h_4$. It has been proved that $h_4 \leq h_5$ and $h_5 \leq h_4$ hence $h_4 = h_5$. This completes the argument for the case $\ast = \wedge$.

If $\ast = \vee$ then h_5 is chosen to be

$$REPR(\lceil A \rceil \sqcup \lceil B \rceil) = REPR(\lceil \overline{A \cup B} \rceil)$$

Since $\ast = \vee$,

$$\mathbf{PI}(h_4) = \mathbf{PI}(REPR(\lceil A \rceil) \vee REPR(\lceil B \rceil)) \cap P_s(\mathbf{X}_n)$$

Obviously $\mathbf{PI}(h_4) \subseteq \mathbf{PI}(REPR(\lceil \overline{A \cup B} \rceil))$ so $h_4 \leq h_5$. It remains to show that $\rho_p(h_4, h_5) \leq 2r^s(sp)^r$.

Let $M_-(f)$ be a random monom in which x_i occurs with probability $1 - p$ and does not occur with probability p . By observing that

$\mathbf{I}(h_4) = \lceil A \rceil \cup \lceil B \rceil$ it is clear that

$$\begin{aligned} \rho_p(h_4, h_5) &= \text{Prob}[M_-(f) \in \delta_-(\lceil A \rceil, \lceil B \rceil)] \\ &\leq 2r^s \text{Gap}(f) \end{aligned}$$

from Lemma(3.18) and Lemma(3.20).

We can produce an upper bound on $\text{Gap}(f)$ with the chosen $M_-(f)$, by adapting the techniques of Lemma(3.26). So it is sufficient to show that if $C \subseteq U(d)$ and $C \vdash E$ then

$$\text{Prob}[M_-(f) \in (\lceil E \rceil - \lceil C \rceil)] \leq (sp)^r$$

Let $\langle E_1, \dots, E_r \rangle \in \{C\}^r$ which yields E and consider the set of monoms $\{F_i : F_i = E_i - E\}$ which are pairwise disjoint and can be assumed to be non-empty, cf. the proof of Lemma(3.26). We now have that

$$\text{Prob}[M_- \in (\lceil E \rceil - \lceil C \rceil)]$$

is at most

$$\text{Prob}[M_- \in \lceil E \rceil \ \& \ \forall i \ M_- \notin \lceil E_i \rceil]$$

which is equal to

$$\text{Prob}[E \subseteq M_- \ \& \ \forall i \ F_i \subseteq M_-]$$

This quantity is at most

$$\begin{aligned} \text{Prob}[\forall i \ F_i \subseteq M_-] &\leq \left(\max_{1 \leq i \leq r} \{\text{Prob}[F_i \subseteq M_-]\} \right)^r \\ &\leq \left(\sum_{x \in F} \text{Prob}[x \notin M_-] \right)^r \end{aligned}$$

(where F is the maximising F_i)

$$\leq (sp)^r$$

The last line follows from the fact that the events $\{x_i \in M_-\}$ are independent.

This completes the proof of the claim made earlier.

The remainder of the proof is identical to that of Lemma(3.30) but making use of the fact that $|h_1 * h_2| \leq (r - 1)^{2s}$. The details are left to the reader. \square

It is immediate from this result that,

Theorem 3.20: If $f \in M_n$ then

$$L_{r,s}^{n,f}(f) \geq \left\{ \frac{\pi_{s+1}(f)}{(r-1)^{2s}}, \frac{\rho_p(1, f) - sp}{2r^s (sp)^r} \right\} \quad \square$$

With this expression, the explicit lower bound obtained by Andreev (1985) is improved to $2^{n^{1/4} - O(1)}$. This is the same as that achieved in Alon and Boppana (1986) for the same function, using the Lattice method directly.

3.6) Relating Monotone and Combinational Complexity

At the start of this chapter we observed that the aim of studying restricted forms of Boolean network is to enable meaningful results on combinational complexity to be derived. One way in which this can be accomplished is by demonstrating that the chosen simplified network form can efficiently simulate combinational networks. Corollary(3.11) showed that networks over any complete basis can be much more efficient than monotone networks. This might seem to invalidate our reasons for examining monotone network complexity since we cannot hope to find any practical simulation using monotone networks which would work for all functions in M_n . However this fact does

not remove the possibility of there being efficient simulations for certain subsets of M_n , and it might still be possible to derive non-trivial lower bounds on unrestricted network size via similar bounds on monotone complexity.

In this section the important results of Berkowitz (1982) are described. In essence these show for any Boolean function $f(\mathbf{X}_n)$ (not necessarily in M_n), there exists a *monotone* Boolean function g , "related to" f , with the property that $\mathbf{C}(f) = \Omega(\mathbf{C}^m(g))$ and $\mathbf{C}(f) = O(n \mathbf{C}^m(g))$. Thus f has "large" combinational complexity if and only if g has "large" monotone complexity. The precise meaning of "related to" will be made clear below.

The remainder of this section falls into two parts: in the first the concepts of standard circuit and a special type of replacement rule called pseudo-complementation are introduced. A result of Dunne (1984a) is proved which exactly characterises valid pseudo-complements. A consequence of this is a method of transforming combinational networks to monotone networks, but one which is not in general efficient. In the second part we examine a class of monotone functions, called slice functions, which were introduced in Berkowitz (1982). For these functions the transformation in the first part is efficient. Subsequent work on the properties of slice functions from Valiant (1986), Wegener (1985, 1986) and Dunne (1984a, 1985b, 1986) is presented here. We conclude with a result generalising Ugolnikov (1987), from Dunne (1987), which offers a different approach to relating monotone and combinational complexity.

3.6.1) Standard Circuits and Pseudo-Complementation

Monotone Boolean networks employ the incomplete basis $\{\wedge, \vee\}$, whereas combinational networks permit any function in B_2 as

a gate operation. Lemma(1.4) allows us to view combinational complexity and Ω -network complexity as equivalent, to within a constant factor, for any complete $\Omega \subseteq B_2$. So in considering relations between monotone and combinational complexity it is sufficiently general to focus on the complete basis $\{\wedge, \vee, \neg\}$.

Definition 3.15: A *standard circuit* is a Boolean network in which the permitted gate operations are $\{\wedge, \vee\} \subset B_2$ and whose input nodes are $\langle x_1, x_2, \dots, x_n, \bar{x}_1, \bar{x}_2, \dots, \bar{x}_n \rangle$. Thus standard circuits correspond to monotone Boolean networks with negated inputs permitted. For any $f \in B_n$, $\mathbf{SC}(f)$ denote the number of \wedge and \vee gates in the smallest standard circuit realising $f(\mathbf{X}_n)$. •

Lemma 3.32: $\forall f \in B_n, \mathbf{SC}(f) \leq c \cdot \mathbf{C}(f)$, where c is some constant.

Proof: From Lemma(1.4) we have that $\mathbf{C}_{\{\wedge, \vee, \neg\}}(f) \leq c_1 \cdot \mathbf{C}(f)$, for some constant c_1 . So given an optimal combinational network T_0 realising f it may be converted to a $\{\wedge, \vee, \neg\}$ -network T_1 , also realising f , T_1 being only a constant factor larger than T_0 . The only way in which T_1 differs from a standard circuit is that the outputs of some gates of T_1 may be negated. Let g be a last gate in T_1 for which at least one of the wires leaving g enters a negation gate. Without loss of generality suppose that the first r wires leaving g , under some ordering, are negated. Here $1 \leq r \leq \phi(g)$. By applying De Morgan's Laws we can rearrange T_1 in the environment of g using the scheme of Figure(3.9)(i) if $op(g) = \wedge$, Figure(3.9)(ii) if $op(g) = \vee$. This pushes the instances of \neg back one level, so repeating this process eventually ensures that negation is applied to the inputs of T_1 only. The final network is thus a standard circuit realising f and since at most one gate is added for each transformation we have that $\mathbf{SC}(f) \leq 2\mathbf{C}_{\{\wedge, \vee, \neg\}}(f)$ proving the lemma. \square .

Figure 3.9

The only way in which standard circuits differ from monotone networks is that the former permit as inputs $\langle \bar{x}_1, \dots, \bar{x}_n \rangle$. Suppose

that given any standard circuit, T , realising $f \in M_n$, we could find a collection $H = \langle h_1, \dots, h_n \rangle$ of monotone functions with the property that replacing the input \bar{x}_i with h_i did not alter the fact that f was computed. In this way $\mathbf{C}^m(f) \leq \mathbf{SC}(f) + \mathbf{C}^m(H)$ so that if $\mathbf{C}^m(H)$ were small enough, i.e. $\leq \varepsilon \cdot \mathbf{C}^m(f)$, for some $0 < \varepsilon < 1$ then any non-trivial lower bound proved on $\mathbf{C}^m(f)$ would perform hold for $\mathbf{SC}(f)$ and thence $\mathbf{C}(f)$. This motivates the following,

Definition 3.16: Let $f \in M_n$. We say that $h \in M_n$ is a *pseudo-complement* for x_i when computing f if h can replace the input \bar{x}_i in any standard circuit realising f . A collection $H = \langle h_1, \dots, h_n \rangle$ is a *pseudo-complement vector* for $f \in M_n$ if h_i is a pseudo-complement for x_i when computing f , for each $1 \leq i \leq n$. •

Theorem 3.21: (Dunne, 1984a) For any $f \in M_n$, h is a pseudo-complement for x_i when computing f if and only if

$$f^{|x_i:=0}(\mathbf{X}_n - \{x_i\}) \leq h(\mathbf{X}_n) \leq f^{|x_i:=1}(\mathbf{X}_n - \{x_i\})$$

Proof: Let $f \in M_n$ and T be any standard circuit computing f . Consider replacing the input \bar{x}_i of T by a new input z to give a new standard circuit T' realising some function $f'(\mathbf{X}_n, z)$. This function may be written as,

$$f'(\mathbf{X}_n, z) = g_{00} \vee x_i g_{01} \vee z g_{10} \vee x_i z g_{11} \tag{3.49}$$

The functions $g_{\alpha\beta}$ satisfying,

$$\forall m \in \mathbf{PI}(g_{\alpha\beta}) (\bar{x}_i)^\alpha (x_i)^\beta m \text{ is a monom computed by } T \text{ and } m \text{ does not depend on } x_i.$$

where $(x)^\gamma \equiv x$ if $\gamma = 1$ and 1 if $\gamma = 0$.

Thus, since $f \in M_n$,

$$f(\mathbf{X}_n) = g_{00} \vee x_i g_{01} \vee \bar{x}_i g_{10} = g_{00} \vee x_i g_{01} \vee g_{10} \tag{3.50}$$

We can now proceed with the proof of the theorem.

Suppose that $h \in M_n$ is a pseudo-complement for x_i when computing f . We must then have that $f'(\mathbf{X}_n, h(\mathbf{X}_n)) \equiv f(\mathbf{X}_n)$. If $f^{[x_i:=0]} \not\leq h(\mathbf{X}_n)$ then there is some $p \in \mathbf{PI}(f^{[x_i:=0]})$ such that $p \not\leq h(\mathbf{X}_n)$. Consider computing f by a standard circuit for which

$$\begin{aligned}\mathbf{PI}(g_{00}) &= \mathbf{PI}(f^{[x_i:=0]}) - \{p\} \\ g_{01} &= f^{[x_i:=1]} \\ g_{10} &= p \\ g_{11} &= 0\end{aligned}$$

Clearly $g_{00} \vee x_i g_{01} \vee \bar{x}_i g_{10} \equiv f$, but $f'(\mathbf{X}_n, h(\mathbf{X}_n)) \neq f(\mathbf{X}_n)$ because $p \in \mathbf{PI}(f)$ and $p \notin \mathbf{PI}(f')$. It follows that $f^{[x_i:=0]} \leq h$. On the other hand, suppose that $h \not\leq f^{[x_i:=1]}$. Then there is some prime implicant p of h which is not an implicant of $f^{[x_i:=1]}$. In this case using a standard circuit for which $g_{00} = f^{[x_i:=0]}$, $g_{01} = f^{[x_i:=1]}$, $g_{10} = 0$ and $g_{11} = p$ leads to a contradiction. So if h is a pseudo-complement then $f^{[x_i:=0]} \leq h \leq f^{[x_i:=1]}$.

Now suppose that h is such that $f^{[x_i:=0]} \leq h \leq f^{[x_i:=1]}$. We claim that h is a pseudo-complement for x_i when computing f . It must be shown that $f'(\mathbf{X}_n, h(\mathbf{X}_n)) = f(\mathbf{X}_n)$ in this case.

$f'(\mathbf{X}_n, h(\mathbf{X}_n)) \leq f(\mathbf{X}_n)$: From (3.49) and (3.50) it need only be proved that $x_i h g_{11} \leq f$. By definition we have,

$$x_i h g_{11} \leq x_i h \leq x_i \wedge f^{[x_i:=1]} \leq f$$

$f(\mathbf{X}_n) \leq f'(\mathbf{X}_n, h(\mathbf{X}_n))$: Again from (3.49) and (3.50) it is sufficient to prove that $g_{10} \leq h g_{10}$, i.e. $g_{10} \leq h$. Now g_{10} does not depend on x_i and $g_{10} \leq f$, so

$$g_{10} \leq f^{[x_i:=0]} \leq h$$

This completes the proof of the theorem. \square

3.6.2) Slice Functions

Theorem(3.21) offers a method of transforming combinational networks realising any $f \in M_n$ into equivalent monotone networks. By definition, $f^{[x_i := 0]} \leq f^{[x_i := 1]}$ for any $f \in M_n$, so the interval of the theorem is always well-defined. However in general this interval does not appear to yield efficient simulations. Berkowitz (1982) gives the first examples of functions with efficiently computable pseudo-complement vectors. These functions, called k -slice functions, are more than just a class of functions with closely related monotone and combinational complexity; as we shall see below they are of importance in assessing the combinational complexity of any Boolean function by concentrating on monotone networks.

Definition 3.17: Let $f(\mathbf{X}_n) \in B_n$ and k be any natural number such that $1 \leq k \leq n$. The k -slice function of f , denoted f_k , is the monotone Boolean function

$$f_k(\mathbf{X}_n) = (f(\mathbf{X}_n) \wedge T_k^n(\mathbf{X}_n)) \vee T_{k+1}^n(\mathbf{X}_n)$$

Note that f_k is 0 for assignments to \mathbf{X}_n in which fewer than k variables are fixed to 1; is 1 for assignments in which more than k variables are set to 1; and is equal to f for assignments which set exactly k variables to 1. •

Theorem 3.22: (Berkowitz, 1982) For all $f \in B_n$ and $1 \leq k \leq n$, $T_k^{n-1}(\mathbf{X}_n - \{x_i\})$ is a pseudo-complement for x_i when computing $f_k(\mathbf{X}_n)$.

Proof: From Thm(3.21) we need only show that,

$$(f_k)^{[x_i := 0]} \leq T_k^{n-1} \leq (f_k)^{[x_i := 1]}$$

$$\begin{aligned}
(f_k)^{x_i := 0} &= (f^{x_i := 0} \wedge T_k^{n-1} \vee T_{k+1}^{n-1})(\mathbf{X}_n - \{x_i\}) \\
&\leq T_k^{n-1}(\mathbf{X}_n - \{x_i\}) \\
&\leq (f^{x_i := 1} \wedge T_{k-1}^{n-1} \vee T_k^{n-1})(\mathbf{X}_n - \{x_i\}) \\
&= (f_k)^{x_i := 1}
\end{aligned}$$

proving the theorem. \square .

From this theorem we have that the collection of functions,

$$H_{n,k} = \langle T_{k-1}^n(\mathbf{X}_n - \{x_1\}), \dots, T_{k-1}^n(\mathbf{X}_n - \{x_n\}) \rangle$$

is a pseudo-complement vector for any k -slice function f_k . The next result, independently obtained by Wegener (1985), Valiant (1986) and Paterson (pers. communication) shows that $H_{n,k}$ can be computed by efficient monotone networks. The bound for k constant was also derived by McColl (pers. communication).

Lemma 3.32: $\mathbf{C}^m(H_{n,k}) = O(n \min \{k, n-k, (\log n)^2\})$

Proof: The cases k and $n-k$ being constant are similar and are left as an exercise. For arbitrary k , not necessarily constant, our description follows that of Valiant (1986).

It is convenient to assume that $n = 2^m$ for some natural number m and that $k \leq 2^{m-1}$. For $k > 2^{m-1}$, we can build a monotone network for $H_{n,k}$ by constructing one for $H_{2n,k}$ and setting the n extra inputs to 0. It is sufficient to construct a network which is correct when *exactly* k inputs are 1. For then, denoting the i 'th output by y_i we can use the fact that

$$T_k^{n-1}(\mathbf{X}_n - \{x_i\}) \equiv y_i \wedge T_k^n(\mathbf{X}_n) \vee T_{k+1}^n(\mathbf{X}_n)$$

The monotone network realising H_k consists of two parts, essential

building blocks in each part are *merging networks*. A merging network takes as input 2 disjoint lists of Boolean values, d_1, \dots, d_r and e_1, \dots, e_r , where these satisfy $d_i \leq d_{i+1}$, $e_i \leq e_{i+1}$ for all $1 \leq i < r$. The network outputs a list of $2r$ Boolean values, being the two input lists combined and sorted into ascending order. Batcher (1968) constructed monotone merging networks of size $O(r \log r)$ and depth $O(\log r)$.

The following terminology is used in the proof. The $H_{n,k}$ network has n inputs \mathbf{X}_n and n outputs y_1, \dots, y_n . It contains $2 \log n$ *merging levels*. The output nodes at each merging level form a single *layer*. Layers are labelled

$$\{i : 0 \leq i < \log n\} \cup \{2 \log n - i : 0 \leq i < \log n\}$$

Layers i and $2 \log n - i$ consist of $\frac{n}{2^i}$ *lists*, each list containing 2^i nodes. A list, A , *spans* a subset, $\text{span}(A)$ of \mathbf{X}_n , consisting of the variables $\{x_j : r 2^i + 1 \leq j \leq (r+1) 2^i\}$, for some $0 \leq r < \frac{n}{2^i}$. The lists at layers 0 and $2 \log n$ contain respectively a single input x_j or output y_j . The span of the list containing y_j is x_j . Finally we say that the *complement* of a list A , denoted \bar{A} , is formed by concatenating all lists, except for A , which are on the same layer as A .

Assignments to the inputs \mathbf{X}_n induce Boolean values in each list. One, self-evident, feature of the construction will be that the values in any list will always be sorted into ascending order.

Lists will be denoted by upper case Roman letters, and their associated layers by $\text{Layer}(\dots)$.

The network is specified by describing how the lists on each layer are formed from lists on preceding layers.

- L1) $Layer(A) = 0$: then $span(A) = \{x_i\}$ and A is just the input x_i .
- L2) $Layer(A) = i$, where $1 \leq i < \log n$: A is the result of merging B and C , these satisfying $Layer(B) = Layer(C) = i - 1$ and $span(A) = span(B) \cup span(C)$.
- L3) $Layer(D) = \log n + 1$: $|span(D)| = \frac{n}{2}$. Let G be the list on layer $\log n - 1$, which is also of size $\frac{n}{2}$, such that $span(G) \cap span(D) = \emptyset$. D is formed by concatenating the last k bits of G with $\frac{n}{2} - k$ 1's.
- L4) $Layer(D) = 2 \log n - i$ and $0 \leq i < \log n - 1$: let E and F be the lists such that, $layer(E) = layer(D) - 1$, $layer(F) = i$ and $span(E) = span(D) \cup span(F)$. The result of merging E and F is a sorted list of $3 \cdot 2^i$ bits. D consists of the middle 2^i bits of this list.
- L5) $Layer(D) = 2 \log n$: D is the output y_j which will correspond to $T_k^{n-1}(\mathbf{X}_n - \{x_j\})$ when exactly k inputs are 1.

It remains to establish the correctness of this construction. Observe that the first $\log n$ layers form a sorting network. If $Layer(A) = i$, for $0 \leq i < \log n$, then A is a sorted list of the values assigned to $span(A)$. Let $\#_\alpha(A)$ denote the number of α 's in A , where $\alpha = 0$ or 1 , under an assignment to \mathbf{X}_n containing exactly k 1's. To prove correctness is sufficient to establish that,

$$Layer(D) = 2 \log n - i \quad \wedge \quad 0 \leq i < \log n$$

implies

$$\#_0(D) = k - \#_1(\bar{D})$$

We prove this by induction on i from $\log n - 1$ down to 0. The

inductive base $i = \log n - 1$ corresponds to layer $\log n + 1$ so (L3) of the construction applies. Let D and G be as in (L3). Since we consider assignments with exactly k 1's it follows that $\#_1(G) \leq k$ and so from (L3) since G is sorted into ascending order,

$$\#_1(D) = \#_1(G) + \left(\frac{n}{2} - k\right)$$

By definition $\text{span}(G) = \bar{D}$, hence $\#_1(G) = \#_1(\bar{D})$. Obviously $\#_0(D) = \frac{n}{2} - \#_1(D)$ and so

$$\#_0(D) = k - \#_1(\bar{D})$$

proving the inductive base.

Now assume the assertion above holds for all values $\geq i$, where $0 < i \leq \log n - 1$. We prove it holds for $i - 1$ also. Here (L4) of the construction applies and D , E and F retain the same interpretation as there. By the inductive hypothesis $\#_0(E) = k - \#_1(\bar{E})$. F is a sorted list of the assignment to $\text{span}(F)$ hence $\#_0(F) = 2^i - \#_1(F)$. It follows that,

$$\#_0(D) = 2^i + k - \#_1(\bar{E}) - \#_1(F) = 2^i + k - \#_1(\bar{D})$$

Now $\text{span}(\bar{D}) = \text{span}(\bar{E}) \cup \text{span}(F)$ so the result of merging E and F consists of $2^i + k - \#_1(\bar{D})$ 0's followed by 1's. There are k 1's in total and $|D| = 2^i$ so $\#_1(\bar{D})$ must be at least $k - 2^i$. Hence $k - \#_1(\bar{D}) \leq 2^i$. It follows that the middle 2^i bits from merging E and F contain exactly $k - \#_1(\bar{D})$ 0's as required.

The correctness of the construction now follows from the fact that when $D = \{y_j\}$, then $\text{span}(\bar{D}) = \mathbf{X}_n - \{x_j\}$. Thus if exactly k inputs are 1 then y_j will become 0 if and only if x_j is one of the true inputs. \square

Theorem 3.23: (Berkowitz, 1982) For any $f \in B_n$:

$$\text{SF1) } \mathbf{C}(f) \leq \sum_{k=1}^n \mathbf{C}(f_k) + O(n).$$

$$\text{SF2) } \mathbf{C}(f_k) \leq \mathbf{C}(f) + O(n).$$

$$\text{SF3) } \mathbf{C}^{\mathbf{m}}(f_k) \leq \mathbf{C}^{\mathbf{m}}(f) + O(n \cdot \min\{k, n-k, \log n\})$$

$$\text{SF4) } \mathbf{C}^{\mathbf{m}}(f_k) = O(\mathbf{C}(f_k)) + \mathbf{C}^{\mathbf{m}}(H_{n,k})$$

Proof: Recall that $E_k^n(\mathbf{X}_n)$ is the function which is 1 if and only if exactly k inputs are 1. It is obvious that

$$E_k^n = T_k^n \overline{T_{k+1}^n}$$

and so,

$$f_k \wedge \overline{T_{k+1}^n} = f \wedge E_k^n \vee 0 = f \wedge E_k^n$$

Since $f \equiv \bigvee_{k=1}^n f E_k^n$ and $\mathbf{C}(T_k^n) = O(n)$ from Thm(2.20) this establishes (SF1).

(SF2) is immediate from the definition of slice function and Thm(2.20). Similarly (SF3) follows from Thm(3.14), for k or $n-k$ being constant, and the $O(n \log n)$ sorting network of (Ajtai et al., 1983) for arbitrary k . (SF4) is just a restatement of Thm(3.22). In combination with Lemma(3.32) this yields,

$$\mathbf{C}^{\mathbf{m}}(f_k) = O(\mathbf{C}(f_k)) + O(n \cdot \min\{k, n-k, (\log n)^2\}) \quad (\text{SF5})$$

□

(SF1) establishes that if $\mathbf{C}(f) = \omega(n^2(\log n)^2)$ then some k -slice function of f has combinational complexity $\omega(n(\log n)^2)$. Conversely from (SF2) if some slice of f has combinational complexity $\omega(n)$ then $\mathbf{C}(f) = \omega(n)$. Combining this with (SF5) we have that any non-

trivial lower bound on $\mathbf{C}^m(f_k)$, where $\min\{k, n - k\} = O(1)$, implies a lower bound of the same order on $\mathbf{C}(f)$; and if $\mathbf{C}^m(f_k) = h(n) = \omega(n(\log n)^2)$ for any slice then $\mathbf{C}(f) = \Omega(h(n))$. Therefore we can deduce non-trivial lower bounds on the combinational complexity of any Boolean function simply by proving a large enough bound on the *monotone* complexity of one of its slices. Furthermore if the combinational complexity of f is large enough, cf (SF1), then there must be some slice function of f which is suitable, i.e has large monotone complexity.

Slice functions constitute an important "partial" simulation of combinational networks by monotone networks. The remainder of this section examines some specific properties of this class of monotone functions. In particular we consider slice functions of some monotone Boolean *NP*-complete predicates, giving results from Wegener (1985) and Dunne (1984a, 1986). We then consider the question of the relative complexities of f_k and f_{k+1} . Finally we use slice functions to prove that a natural class of monotone functions have equal combinational and monotone network complexities.

The following encodings of three basic *NP*-complete predicates are used.

$\mathbf{X}_n^U = \{x_{ij} : 1 \leq i < j \leq n\}$ and $G(\mathbf{X}_n^U)$ is an undirected n -vertex graph.

$$\frac{n}{2} - \text{clique}(\mathbf{X}_n^U) = \begin{cases} 1 & \text{if } G(\mathbf{X}_n^U) \text{ contains a clique of size } n/2 \\ 0 & \text{otherwise} \end{cases}$$

$$\text{UHC}(\mathbf{X}_n^U) = \begin{cases} 1 & \text{if } G(\mathbf{X}_n^U) \text{ contains a Hamiltonian circuit} \\ 0 & \text{otherwise} \end{cases}$$

Note that $\text{UHC}(\mathbf{X}_n^U)$ may also written as,

$$UHC(\mathbf{X}_n^U) = \bigvee_{\sigma \in S_n} \bigwedge_{i=1}^{n-1} x_{\sigma(i) \sigma(i+1)} \wedge x_{\sigma(n) \sigma(1)}$$

The final function we look at is a more general form of the encoding of satisfiability described in Chapter(2).

$$\begin{aligned} \mathbf{X}_{n,m} &= \{x_{ij} : 1 \leq i \leq n, 1 \leq j \leq m\} \\ \mathbf{Y}_{n,m} &= \{y_{ij} : 1 \leq i \leq n, 1 \leq j \leq m\} \end{aligned}$$

are disjoint sets of Boolean variables. For assignments α to $\mathbf{X}_{n,m}$ and β to $\mathbf{Y}_{n,m}$, $R(\alpha, \beta)$ is the m -clause CNF over $\mathbf{Z} = \{z_1, \dots, z_n\}$ defined by:

$$R(\alpha, \beta) = \bigwedge_{i=1}^m \bigvee_{j=1}^n (\alpha_{ij} z_j \vee \beta_{ij} \bar{z}_j)$$

Thus the literal z_j occurs in the i 'th clause if and only if $x_{ij} = 1$ under α ; the literal \bar{z}_j occurs in the i 'th clause if and only if $y_{ij} = 1$ under β . $SAT(\mathbf{X}_{n,m}, \mathbf{Y}_{n,m})$ is the (monotone) Boolean function which equals 1 if and only if the CNF $R(\mathbf{X}_{n,m}, \mathbf{Y}_{n,m})$ is satisfiable.

Suppose g is any one of these 3 functions. Then any two distinct prime implicants of g contain exactly the same number of variables. Thus if $g \equiv \frac{n}{2}$ -clique, then each prime implicant of g depends on $\frac{n}{2} \binom{n}{2} - 1$ variables, being the number of edges in such a clique; if $g \equiv UHC$ then each prime implicant of g contains exactly n variables; finally if $g \equiv SAT(\mathbf{X}_{n,m}, \mathbf{Y}_{n,m})$ then each prime implicant of g contains exactly m variables. Any $f \in M_n$ such that for all $p \in \mathbf{PI}(f)$, $|\text{var}(p)| = k$ is said to be k -homogeneous. $Q_{n,k}$ denotes the set of k -homogeneous functions in M_n .

We will be particularly concerned with the following specific slice functions.

Definition 3.18: If $f \in Q_{n,k}$ then the *canonical slice* of f , denoted $c - sl(f)$ is its k -slice function.

For any $f \in M_n$, the *central slice* of f , denoted $Cen(f)$, is its $\frac{n}{2}$ -slice function. •

For any $f \in Q_{n,k}$ it should be clear that $f \wedge T_k^n \equiv f$ and thus $c - sl(f) = f \vee T_{k+1}^n$. The canonical slice appears to be very similar to f and so for the three functions given above it would seem to be a natural candidate to examine as a potential "hard" slice function.

Lemma 3.33: Let $N = \frac{n}{2} = |\mathbf{X}_n^U|$.

- i) $C^m(c - sl(\frac{n}{2} - clique)) = O(N \log N)$ (Wegener, 1985)
- ii) $C^m(c - sl(UHC)) = O(N^2)$ (Dunne, 1986)
- iii) $C^m(c - sl(SAT)) = O(nm (\log nm)^2)$ (Dunne, 1986)

Proof: Let f be any of the three functions above and k its canonical slice. By definition, $f_k = f \wedge T_k^n \vee T_{k+1}^n$. Since f_k is 0 (resp. 1) whenever less than (resp. more than) k inputs are 1, and equal to f whenever exactly k inputs equal 1, so for any function g which is equal to f , for assignments containing exactly k 1s we have $g_k = f_k$. Thus instead of using f we may substitute any function, g , which agrees with f on inputs having exactly k 1s.

- i) Let $k = \frac{n}{2} \cdot (\frac{n}{2} - 1)$ and choose g to be the function,

$$T_k^n (T_{k-1}^{n-1}(\mathbf{X}^{(1)}), \dots, T_{k-1}^{n-1}(\mathbf{X}^{(n)}))$$

where, $\mathbf{X}^{(i)} = \{x_{ji} : 1 \leq j \neq i \leq n\}$.

Clearly this function can be computed with $O(N \log N)$ monotone gates. That the substitution is correct follows from the easily established fact

An n -vertex graph with exactly k edges contains an $\frac{n}{2}$ -clique if and only if at least $\frac{n}{2}$ vertices have degree at least $\frac{n}{2} - 1$.

ii) Let $UCON(\mathbf{X}_n^U)$ be the monotone Boolean function which is true if and only if $G(\mathbf{X}_n^U)$ is connected. By using a transitive closure algorithm, applied to the adjacency matrix of $G(\mathbf{X}_n^U)$ it follows that $C^m(UCON(\mathbf{X}_n^U)) = O(N^2)$. Using $\mathbf{X}^{(i)}$ as in (i) we choose the substituting function g to be,

$$g(\mathbf{X}_n^U) = \bigwedge_{i=1}^n T_2^{n-1}(\mathbf{X}^{(i)}) \wedge UCON(\mathbf{X}_n^U)$$

That this is correct follows from the fact that

An n -vertex graph with exactly n edges contains a Hamiltonian circuit if and only if every vertex has degree at least 2 and the graph is connected.

iii) Let,

$$\begin{aligned} CL_i &= \{x_{1i}, \dots, x_{ni}, y_{1i}, \dots, y_{ni}\} \\ ZP_i &= \{x_{i1}, \dots, x_{im}\} \\ ZN_i &= \{y_{i1}, \dots, y_{im}\} \end{aligned}$$

The substituting function $g(\mathbf{X}_{n,m}, \mathbf{Y}_{n,m})$, agreeing with $SAT(\mathbf{X}_{n,m}, \mathbf{Y}_{n,m})$ when exactly m inputs are 1 is given by,

$$\bigwedge_{i=1}^m T_1^{2n}(CL_i) \wedge \overline{\left(\bigwedge_{i=1}^n T_1^m(ZP_i) \wedge T_1^m(ZN_i) \right)}$$

This expression encodes the condition that each of the m clauses of the CNF $R(\mathbf{X}_{n,m}, \mathbf{Y}_{n,m})$ contains at least one literal and at most one of the clauses $(z_i), (\bar{z}_i)$ occur in R for each $1 \leq i \leq n$. It is easy to see that any m -clause CNF containing exactly m literals is satisfiable if

and only if it meets this condition. The upper bound on $C^m(c - sl(SAT))$ now follows since the instances of negation, used in the definition of $g(\mathbf{X}_{n,m}, \mathbf{Y}_{n,m})$ can be eliminated using (SF5) of Thm(3.23). \square

It is reasonable to conjecture that all the NP -complete problems above require exponential size combinational networks and so we expect each of them to have some hard slice function. Lemma(3.33) has shown that the canonical slice is not a suitable candidate. The following theorem does identify a specific slice function which is "probably" hard, in the sense that the corresponding decision problem is NP -complete.

Theorem 3.24: (Dunne, 1986)

- i) $Cen(\frac{n}{2} - clique)$ is NP -complete.
- ii) $Cen(UHC)$ is NP -complete.
- iii) $Cen(SAT)$ is NP -complete.

Proof: All three results involve constructing a projection from the central slice of a larger instance of g onto g itself, thus in terms of Defn(2.1), g is a p -projection of $Cen(g)$; here g is one of the functions in the theorem statement. The projection is constructed so that exactly half of the arguments are 1 for any assignment. We use $e(n)$ to denote the value $\frac{n}{2}$. In the proof of (i) and (ii) we assume without loss of generality that n is an exact multiple of 4.

i) Any assignment to \mathbf{X}_n^U defines some n -vertex graph G . Given any n -vertex graph G we construct a $5n$ -vertex graph H with the following properties:

- P1) H contains a $\frac{5n}{2}$ -clique if and only if G contains an $\frac{n}{2}$ -clique.

P2) H has exactly $\frac{e(5n)}{2}$ edges.

Clearly if H can be constructed then we can compute $\frac{n}{2} - \text{clique}(\mathbf{X}_n^U)$ since the method of constructing H using $G(\mathbf{X}_n^U)$ defines a p -projection from $\text{Cen}(\frac{5n}{2} - \text{clique})$ onto $\frac{n}{2} - \text{clique}$.

Given G , the $5n$ -vertex graph H is constructed as follows:

H consists of 3 graphs: G with vertices $\{v_1, \dots, v_n\}$; \bar{G} with vertices $\{u_1, \dots, u_n\}$ and G^* with vertex set $\{w_1, \dots, w_{3n}\}$. \bar{G} is the complement of G with respect to K_n , i.e the graph such that

$$\{u_i, u_j\} \in E(\bar{G}) \iff \{v_i, v_j\} \notin E(G)$$

The vertices $\{w_1, \dots, w_{2n}\}$ form a $2n$ -clique in G^* . Additionally G^* contains $\frac{7n^2 + n}{4}$ edges not in this clique, but does not have a $(2n + 1)$ -clique. Finally there are edges

$$\{w_i, v_j\} \quad \forall \quad 1 \leq i \leq 2n, \quad 1 \leq j \leq n$$

That G^* can be constructed is an easy consequence of Turán's Theorem, see e.g (Berge, p. 280).

Obviously if G contains an $\frac{n}{2}$ -clique then in conjunction with $\{w_1, \dots, w_{2n}\}$ this gives a $\frac{5n}{2}$ -clique in H . On the other hand if H contains a $\frac{5n}{2}$ -clique then this cannot involve any vertices of \bar{G} , which is not connected to G or G^* , and can contain at most $2n$ -vertices of G^* , since this does not contain a $(2n + 1)$ -clique. It follows that any $\frac{5n}{2}$ -clique in H uses at least $\frac{n}{2}$ vertices of G and hence G contains an $\frac{n}{2}$ -clique. Thus property (P1) holds of H .

Counting the number of edges in H yields;

$$\begin{aligned} |E(H)| &= |E(G)| + |E(\bar{G})| + |E(G^*)| + 2n^2 \\ &= e(n) + e(2n) + \frac{7n^2 + n}{4} + 2n^2 = \frac{e(5n)}{2} \end{aligned}$$

So property (P2) holds also and part (i) of the theorem follows.

ii) In the same manner as (i) given G , an n -vertex graph, we construct H , a $7n$ -vertex graph, for which,

P1) H contains a Hamiltonian circuit if and only if G contains a Hamiltonian circuit.

P2) H has exactly $\frac{e(7n)}{2}$ edges.

As before this implies that UHC is a p -projection of $Cen(UHC)$. H is constructed as follows from a given G .

H again consists of 3 graphs: G with vertices $\{v_1, \dots, v_n\}$; \bar{G} with vertices $\{u_1, \dots, u_n\}$; and G^* with vertices $\{w_1, \dots, w_{5n}\}$. \bar{G} is the complement of G with respect to K_n . H contains also the following edges,

$$\left\{ \begin{array}{l} \{v_1, u_1\} \\ \{u_i, w_i\} \quad \forall 1 \leq i \leq n \\ \{w_i, u_{i+1}\} \quad \forall 1 \leq i < n \\ \{w_i, w_{i+1}\} \quad \forall n \leq i < 5n \\ \{w_{5n}, v_i\} \quad \forall v_i \in \Gamma(v_1) \\ \{w_{5n}, u_i\} \quad \forall u_i \in \Gamma(w_1) \end{array} \right\}$$

Here $\Gamma(x)$ is the set of vertices adjacent to x in G , if $x = v_1$, or in \bar{G} if $x = u_1$.

In addition to these G^* contain an extra $\beta(n) = \frac{47n^2 - 33n - 4}{4}$ edges. It should be clear that H has a path from v_1 to w_{5n} , which path contains all the vertices in \bar{G} and G^* and so has $6n$ edges. From the construction of H we have,

$$|E(H)| = |E(G)| + |E(\bar{G})| + |\text{Edges in path } v_1 \text{ to } w_{5n}| + n - 1 + \beta(n)$$

The $n - 1$ term is the total number of edges added between w_{5n} and vertices adjacent to v_1 or w_1 .

$$= e(n) + 7n - 1 + \beta(n) = \frac{e(7n)}{2}$$

So H contains the correct number of edges. It remains to establish that H has a Hamiltonian circuit if and only if G does.

Suppose G contains a Hamiltonian circuit,

$$v_1 \longleftrightarrow x \in \Gamma(v_1) \longleftrightarrow \dots \longleftrightarrow y \in \Gamma(v_1) \longleftrightarrow v_1$$

By the construction there is a Hamiltonian path, HP , joining all the vertices $\{v_1, V(\bar{G}), V(G^*)\}$, which path commences at v_1 and terminates in w_{5n} . Thus,

$$v_1 \longleftrightarrow HP \longleftrightarrow w_{5n} \longleftrightarrow \dots \longleftrightarrow y \longleftrightarrow v_1$$

is a Hamiltonian circuit in H . On the other hand suppose that H contains a Hamiltonian circuit. We claim that there are exactly two edges in this circuit which connect a vertex of G to any vertex of \bar{G} or G^* , and that one of these edges is $\{v_1, u_1\}$; the other being $\{w_{5n}, v_j\}$ for some $v_j \in \Gamma(v_1)$. Obviously there must be at least two edges between $V(G)$ and the other vertices in H . Since there are exactly two vertices in $V(\bar{G}) \cup V(G^*)$ adjacent to vertices of G and each vertex can occur only once in a Hamiltonian circuit this assertion is immediate. Thus any Hamiltonian circuit C in H must be of the form,

$$\begin{aligned}
 v_1 &\leftrightarrow u_1 \leftrightarrow c_1 \leftrightarrow \dots \\
 \dots &\leftrightarrow c_{6n-1} \leftrightarrow w_{5n} \leftrightarrow x \leftrightarrow d_3 \leftrightarrow \dots \\
 \dots &\leftrightarrow d_n \leftrightarrow y \leftrightarrow v_1
 \end{aligned}$$

where $c_i \in V(\bar{G}) \cup V(G^*)$, $d_j \in V(G)$ and $x, y \in \Gamma(v_1)$.

It follows that

$$v_1 \leftrightarrow x \leftrightarrow d_3 \leftrightarrow \dots \leftrightarrow d_n \leftrightarrow y \leftrightarrow v_1$$

is a Hamiltonian circuit in G . This establishes part (ii) of the Theorem.

(iii) For this case, we construct a p -projection from $Cen(SAT)$ onto SAT by forming a $3m$ clause CNF Q over the literal set $\mathbf{Z}_n \cup \mathbf{U}_n$ being,

$$\{z_1, \dots, z_n, \bar{z}_1, \dots, \bar{z}_n, u_1, \dots, u_n, \bar{u}_1, \dots, \bar{u}_n\}$$

Q is defined using the m -clause CNF $P \equiv R(\mathbf{X}_{n,m}, \mathbf{Y}_{n,m})$ in such a way that Q contains exactly $6nm$ literals (i.e half the number of possible literals) and is satisfiable if and only if P is.

Given P , which is an m -clause CNF over the literal set \mathbf{Z}_n , Q is given by,

$$Q(\mathbf{Z}_n, \mathbf{U}_n) = P(\mathbf{Z}_n) \wedge Pcomp(\mathbf{Z}_n, \mathbf{U}_n) \wedge P^*(\mathbf{U}_n)$$

$Pcomp$ consists of m clauses; the i 'th clause contains all the literals in \mathbf{U}_n and additionally all the literals which do not occur in the i 'th clause of P ; P^* also consists of m clauses each of which contains all the literals of \mathbf{U}_n . Now if P is satisfiable, then certainly Q is; simply set $u_1 = 1$. If Q is satisfiable, then since P does not depend on the literals \mathbf{U}_n , any assignment which satisfies Q must satisfy P also. So P is satisfiable if and only if Q is. It is immediate from the construction that Q contains exactly $6nm$ literals and this establishes (iii) and

the theorem. \square

It is a trivial matter to generalise this to,

Corollary 3.15: $\forall 0 < \varepsilon < 1$, if $g \in \{\frac{n}{2} - \text{clique}, \text{UHC}, \text{SAT}\}$ then the εN -slice of g is *NP*-complete, N being the number of inputs of g . \square

We saw in Lemma(3.33) that the canonical slice, which is superficially that slice function most similar to f , may be easy to compute. In contrast, Thm(3.24) presents some evidence that the central slice is likely to be of superpolynomial complexity for some specific functions. The next result considers the relation between $\mathbf{C}^{\mathbf{m}}(f_k)$ and $\mathbf{C}^{\mathbf{m}}(f_{k+1})$.

Theorem 3.25: Part (i): (Dunne, 1986); Part (ii) (Wegener, 1986).

i) Let $f \in M_n$ such that $c - \text{sl}(f)$ exists and is the k -slice f_k . $\forall c \geq 1$, $\mathbf{C}^{\mathbf{m}}(f_{k+c})$ is at most

$$n^2 + 1 + \mathbf{C}^{\mathbf{m}}(T_{k+c}^n) + \mathbf{C}^{\mathbf{m}}(T_{k+c+1}^n) + n\mathbf{C}^{\mathbf{m}}(f_{k+c-1})$$

ii) Let

$$l(k, n) = \frac{\binom{n-1}{k-1}}{\log \binom{n-1}{k-1}}$$

There exist monotone Boolean functions, $f \in M_n$, for which the canonical slice is the k -slice f_k and such that,

$$\mathbf{C}^{\mathbf{m}}(f_k) \geq \mathbf{C}(f_k) = \Omega(l(k, n))$$

but

$$\mathbf{C}^{\mathbf{m}}(f_{k+c}) = O(n \log n) \quad \forall c \geq 1$$

Proof: For (i) it is sufficient to construct a suitable substituting function for f , in the same way as Lemma(3.33). Let $h_i : \mathbf{X}_n \rightarrow \{0, 1\}^n$ be given by,

$$\begin{cases} \{x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n\} & \text{if } x_i = 1 \\ \{0, 0, 0, \dots, 0, 0\} & \text{otherwise} \end{cases}$$

We then have that $f_{k+c}(\mathbf{X}_n)$ is

$$\left(\left(\bigvee_{i=1}^n f_{k+c-1}(h_i(\mathbf{X}_n)) \right) \wedge T_{k+c}^n(\mathbf{X}_n) \right) \vee T_{k+c+1}^n(\mathbf{X}_n)$$

i.e $f(\mathbf{X}_n)$ is 1 when exactly $k+c$ inputs are true if and only if for some $k+c-1$ size subset of the true inputs, f is 1 when exactly these $k+c-1$ inputs are true. This holds because every prime implicant of f contains exactly k variables. Part (i) now follows since,

$$h_i(\mathbf{X}_n) = \{x_1 x_i, \dots, x_{i-1} x_i, 0, x_{i+1} x_i, \dots, x_n x_i\}$$

which has monotone complexity $n-1$.

For (ii) let $S_{k,n}$ be the set of k -slice functions, f_k such that all monoms of length k which do not depend on x_1 are prime implicants of f . Clearly there are $\binom{n-1}{k-1}$ monoms of length k which do contain x_1 and there is a bijective mapping between subsets of these and functions in $S_{k,n}$. Corollary(2.2) establishes that almost all functions in $S_{k,n}$ have combinational complexity $\Omega(l(k,n))$ proving the lower bound. Now consider f_{k+c} for any $c > 1$. From the definition of slice function it is sufficient to establish that $f_{k+c} \equiv T_{k+c}^n$. Now certainly if fewer than $k+c$ inputs are true then both of these function are 0. If at least $k+c$ are true then there is a subset $\{x_{i_1}, \dots, x_{i_k}\}$ of these true inputs which does not contain x_1 . From the definition of f ,

$x_{i_1} \cdots x_{i_k} \leq f$ and so both expressions are equal in this case also. The previously established upper bounds on T_k^n now complete the proof. \square

At present it is not known if the techniques of Section(3.5) can be made to work for slice functions. One way in which we might simplify the process of deriving non-trivial lower bounds on combinational complexity is by finding other classes of monotone functions with asymptotically equal monotone and combinational complexity. Dunne (1985b) shows that "almost all" functions in $Q_{n,n-k}$, for k constant, form such a class.

Let $\mathbf{C}^{\mathbf{m}^*}(f)$ denote the minimal number of monotone gates in a network realising f with inputs $\mathbf{X}_n \cup \{f_1, \dots, f_n\}$, and \mathbf{C}^* the analogous measure for combinational complexity. We know from Thm(3.23) (SF1) that $\mathbf{C}^*(f) = O(n)$ for all $f \in B_n$. From the same theorem it also follows that

$$\mathbf{C}^{\mathbf{m}}(f) \leq \mathbf{C}(f) + O((n \log n)^2) + \mathbf{C}^{\mathbf{m}^*}(f)$$

So for any $f \in M_n$ for which we can prove $\mathbf{C}^{\mathbf{m}^*} < \varepsilon \mathbf{C}^{\mathbf{m}}(f)$, for some $0 < \varepsilon \leq 1$ we have $\mathbf{C}(f) = \Omega(\mathbf{C}^{\mathbf{m}}(f))$, provided that $\mathbf{C}^{\mathbf{m}}(f)$ is large enough.

Lemma 3.33: Let $k \geq 1$ be constant and $Q_{n,k}^m$ denote the set of n -input m output Boolean functions such that for each $F = \langle f^1, \dots, f^m \rangle \in Q_{n,k}^m$, we have $f^j \in Q_{n,k}$ for all $1 \leq j \leq m$. As before let $\mathbf{C}^{\mathbf{m}}(Q_{n,k}^m)$ denote the maximal monotone complexity of any function in $Q_{n,k}^m$. Then,

- i) $\mathbf{C}^{\mathbf{m}}(Q_{n,1}^n) = O(n^2 / \log n)$
- ii) $\mathbf{C}^{\mathbf{m}}(Q_{n,k}^n) \leq n \mathbf{C}^{\mathbf{m}}(Q_{n,k})$

- iii) $\mathbf{C}^{\mathbf{m}}(Q_{n,k}) \leq \mathbf{C}^{\mathbf{m}}(Q_{n,k-1}) + 2n - 1$
 iv) For $k \geq 2$, $\mathbf{C}^{\mathbf{m}}(Q_{n,k}) = O(n^k / \log n)$.

Proof: (Details omitted) (i) is given by Savage (1974) and (ii) is obvious. (iii) is from Wegener (1987) and (iv) immediate from (i-iii). \square

Theorem 3.26: (Dunne, 1985b) Let $k \in \mathbf{N}$ and $Q_{n,n-k} \subset M_n$ the class of $(n-k)$ -homogeneous functions. Then $\forall f \in Q_{n,n-k}$ $\mathbf{C}^{\mathbf{m}^*}(f) = O(n^{k-1} / \log n)$.

Proof: Let $f \in Q_{n,n-k}$. Clearly

$$f = f \wedge T_{n-k}^n = f \wedge T_{n-k}^n \vee T_n^n$$

So it is sufficient to prove that for all $2 \leq q \leq k$

$$\mathbf{C}^{\mathbf{m}}(f \vee T_{n-k+q}^n) \leq \mathbf{C}^{\mathbf{m}}(f \vee T_{n-k+q-1}^n) + O\left(\frac{n^{k-1}}{\log n}\right) \quad (3.51)$$

As a result of this it will follow that f can be computed from its canonical slice function, f_{n-k} using only $O(n^{k-1} / \log n)$ extra gates. Since f_{n-k} is counted at no cost in the measure $\mathbf{C}^{\mathbf{m}^*}$ this proves the theorem.

We shall actually prove a slightly stronger result than (3.51), namely $\forall 2 \leq q \leq k$,

$$\mathbf{C}^{\mathbf{m}}(f \vee T_{n-k+q}^n) \leq \mathbf{C}^{\mathbf{m}}(f \vee T_{n-k+q-1}^n) + O\left(\frac{n^{k-q+1}}{\log n}\right) \quad (3.52)$$

Let S_{q-1} be an optimal monotone network realising $f \vee T_{n-k+q-1}^n$. This function may be written as,

$$f \vee p_1 \vee p_2 \vee \cdots \vee p_t$$

where for each p_i : $p_i \preceq f$.

Recall that for any monom p , $\chi(p)$ is the disjunction over all variables of \mathbf{X}_n which do not occur in $\text{var}(p)$. We claim that for all prime implicants m of f , and all p_i it holds that $m \leq \chi(p_i)$. This assertion follows easily from the fact that $m \not\leq p_i$ hence there is some $x \in \mathbf{X}_n$ such that $m \leq x \leq \chi(p_i)$. Let S_q be the network which computes,

$$(f \vee T_{n-k+q-1}^n) \wedge \bigwedge_{i=1}^t \chi(p_i)$$

From the preceding argument S_q realises $f \vee T_{n-k+q}^n$.

$\bigwedge_{i=1}^t \chi(p_i)$ is the dual of a $k - q + 1$ -homogeneous function, and so from Lemma(3.33)(iv) this can be computed by a monotone network containing $O(n^{k-q+1}/\log n)$ gates. This proves (B) and the theorem. \square

Corollary 3.16: If $f \in \mathcal{Q}_{n,n-k}$, for which $\mathbf{C}^m(f) = \omega(n^{k-1}/\log n)$ then

$$\mathbf{C}(f) = \Omega(\mathbf{C}^m(f))$$

Proof: From the theorem it follows that

$$\mathbf{C}^m(f) \leq \mathbf{C}(f) + O(n) + O(n^{k-1}/\log n)$$

Noting that $|\mathcal{Q}_{n,n-k}| = 2^{\Omega(n^k)}$, from Corollary(2.2) we have that almost all $f \in \mathcal{Q}_{n,n-k}$ have combinational complexity $\Omega(n^k/\log n)$ thus almost all functions in $\mathcal{Q}_{n,n-k}$ have monotone complexity $\omega(n^{k-1}/\log n)$. \square

In contrast to this result we have,

Theorem 3.27: (Dunne, 1985b)

i) Let n be even and \mathbf{X}_n be partitioned into $n/2$ sets $\mathbf{X}_n^{(i)}$ defined by $\mathbf{X}_n^{(i)} = \{x_i, x_{\frac{n}{2}+i}\}$ for $1 \leq i \leq n/2$. Let $J_n \subset M_n$ such that for all $f \in J_n$ each prime implicant of f contains exactly one variable from each

partition class $\mathbf{X}_n^{(i)}$. Then for almost all $f \in J_n$

$$\mathbf{C}^{\mathbf{m}^*}(f) = \Omega\left(\frac{2^{n/2}}{n}\right); \mathbf{C}^{\mathbf{m}}(f) = \Omega\left(\frac{2^{n/2}}{n}\right)$$

ii) There exist functions $f \in \mathcal{Q}_{n,n-k}$, for k constant, such that

$$\mathbf{C}^{\mathbf{m}^*}(f) = \Omega\left(\frac{n^{k-1}}{\log n}\right)$$

Proof: Omitted. \square

Ugolnikov (1987) uses a different approach to the problem of relating monotone and non-monotone bases. Consider the 3 bases

$$\{\mu\} ; \{\mu, \oplus\} ; \{\mu, \oplus, 0\}$$

where μ is the 3-input majority function.

The last basis is logically complete and the first realises exactly the class of *self-dual* monotone Boolean functions.

Ugolnikov proved that for any Boolean function, f , computable in the basis $\{\mu, \oplus\}$ there existed a self-dual monotone Boolean function, g , such that $g \leq f$ and for which

$$\mathbf{C}_\mu(g) \leq \mathbf{C}_{\{\mu, \oplus\}}(f) \tag{3.52}$$

and furthermore, for any such f and any self-dual monotone function, g such that $g \leq f$ it holds,

$$\mathbf{C}_{\{\mu, \oplus\}}(f) \leq \mathbf{C}_{\{\mu, \oplus, 0\}}(f) + \mathbf{C}_\mu(g) + 2 \tag{3.53}$$

Given this result suppose that $f(\mathbf{X}_n)$ is a monotone Boolean function of n variables $\mathbf{X}_n = \langle x_1, \dots, x_n \rangle$. It is easy to see that the function $F(y, z, \mathbf{X}_n)$ defined by

$$F(y, z, \mathbf{X}_n) = y \wedge (z \vee f(\mathbf{X}_n)) \vee z \wedge \tilde{f}(\mathbf{X}_n) \tag{3.54}$$

is self-dual and that $\mathbf{C}^m(f) \leq 4 \mathbf{C}_\mu(F)$. Now since F is self-dual it follows that the only self-dual function g such that $g \leq F$ is F itself. Hence,

$$\mathbf{C}_{\{\mu, \varnothing\}}(F) = \mathbf{C}_\mu(F) \geq \frac{\mathbf{C}_{\{\wedge, \vee, 0, 1\}}(f)}{4} \quad (3.55)$$

(3.55) is the form stated in Ugolnikov (1987) and as such does not give a direct relationship between the monotone complexity of f and the combinational complexity of F . However such a relation may be obtained by widening the scope of g in (3.53); thus if $g \leq f$ and g is computable by the basis $\{\mu, \varnothing\}$ then

$$\mathbf{C}_{\{\mu, \varnothing\}}(f) \leq \mathbf{C}_{\{\mu, \varnothing, 0\}}(f) + \mathbf{C}_{\{\mu, \varnothing\}}(g) + 2 \quad (3.56)$$

This, combined with (3.55), gives a lower bound on the combinational complexity of F in terms of the monotone complexity of f and an upper bound on the $\{\mu, \varnothing\}$ complexity of g .

In this section we generalise inequalities (3.52) and (3.53) to other non-monotone and complete bases. Ugolnikov's results connect a monotone basis Ω_1 ($\{\mu\}$), an extension of Ω_1 by a basis of non-monotone functions Ω_2 ($\{\varnothing\}$), and an extension of the incomplete basis $\Omega_1 \cup \Omega_2$ by a constant function α ($\{0\}$) to a complete basis. Using Ugolnikov's proof of (3.52) and (3.53) as a foundation we establish sufficient conditions on $\langle \Omega_1, \Omega_2, \alpha \rangle$ which allow analogues of the inequalities (3.52) and (3.53) to be derived. Below we introduce notation used subsequently.

For any basis $\Omega \subset \bigcup_{i=0}^k B_k$ of constant arity Boolean functions, $[\Omega]$ will denote the set of functions which can be computed by networks over the basis Ω . For $\alpha \in \{0, 1\}$ the relation \leq_α is defined over $f, g \in B_n$ by,

$$g \leq_{\alpha} f \iff \begin{cases} g \leq f & \text{for } \alpha = 0 \\ f \leq g & \text{for } \alpha = 1 \end{cases}$$

Finally $\theta_{\alpha} \in M_2$ if the function $x \vee y$ if $\alpha = 0$, $x \wedge y$ if $\alpha = 1$.

Definition 3.19: Let Ω_1 and Ω_2 be disjoint bases and $\alpha \in \{0, 1\}$. $\langle \Omega_1, \Omega_2, \alpha \rangle$ is *sympathetic* if and only if each of the following conditions holds.

- i) Ω_1 is monotone and contains a non-constant function of at least two arguments.
- ii) Ω_2 contains only non-monotone functions.
- iii) The basis $\Omega_1 \cup \Omega_2$ is not complete.
- iv) The basis $\Omega_1 \cup \Omega_2 \cup \{\alpha\}$ is complete.
- v) $\forall \phi \in \Omega_2, \exists \psi \in [\Omega_1]$ such that $\mathbf{C}_{\Omega_1}(\psi) \leq 1$ and for which $\psi \leq_{\alpha} \phi$.
- vi) $\theta_{\alpha} \in [\Omega_1 \cup \Omega_2]$. •

Theorem 3.28: (Dunne, 1987) extending (Ugolnikov, 1987) If $\langle \Omega_1, \Omega_2, \alpha \rangle$ is sympathetic then for all $f \in [\Omega_1 \cup \Omega_2]$,

- a) There exists $g \in [\Omega_1]$ such that $g \leq_{\alpha} f$ and for which

$$\mathbf{C}_{\Omega_1}(g) \leq \mathbf{C}_{\Omega_1 \cup \Omega_2}(f)$$

- b) For all $g \in [\Omega_1]$ such that $g \leq_{\alpha} f$

$$\mathbf{C}_{\Omega_1 \cup \Omega_2}(f) \leq \mathbf{C}_{\Omega_1 \cup \Omega_2 \cup \{\alpha\}}(f) + \mathbf{C}_{\Omega_1}(g) + \mathbf{C}_{\Omega_1 \cup \Omega_2}(\theta_{\alpha})$$

- c) For all $g \in [\Omega_1 \cup \Omega_2]$ such that $g \leq_{\alpha} f$

$$\mathbf{C}_{\Omega_1 \cup \Omega_2}(f) \leq \mathbf{C}_{\Omega_1 \cup \Omega_2 \cup \{\alpha\}}(f) + \mathbf{C}_{\Omega_1 \cup \Omega_2}(g) + \mathbf{C}_{\Omega_1 \cup \Omega_2}(\theta_{\alpha})$$

Proof: (b) and (c) employ identical arguments so only the stronger result (c) is given in detail.

a) Let $f \in [\Omega_1 \cup \Omega_2]$ and S a network over the basis $\Omega_1 \cup \Omega_2$ realising f . Further let q denote the number of gates in S whose operation is a function in Ω_2 . We show that for all $q \geq 0$, if S is a $\Omega_1 \cup \Omega_2$ network realising some function f and using q Ω_2 gates, then there is a network S_1 realising a function f_1 , such that $f_1 \leq_\alpha f$, and which uses at most $\max\{0, q-1\}$ Ω_2 gates. Clearly this is sufficient to prove (a).

If $q = 0$, then the result is trivial since $f \in [\Omega_1]$, so choosing $g = f$ gives the bound in this case. So suppose the assertion of the preceding paragraph is true of all appropriate networks containing at most $q-1$ Ω_2 gates and let S be a network realising f and using q Ω_2 gates. Let v be a "last" Ω_2 gate in S , i.e a gate all of whose descendants are Ω_1 gates. Let v_1, v_2, \dots, v_k denote the input gates of v , these computing functions h_1, h_2, \dots, h_k in $[\Omega_1 \cup \Omega_2]$. Furthermore let $\phi \in \Omega_2$ denote the operation of v , so that v computes a function $h = \phi(h_1, h_2, \dots, h_k)$. Now from (v) of the definition of sympathetic there is some function $\psi \in [\Omega_1]$ such that $C_{\Omega_1}(\psi) \leq 1$ and for which $\psi \leq_\alpha \phi$. Let S_1 be the network obtained from S by replacing the ϕ -gate v with $\psi(h_1, \dots, h_k)$ and f_1 be the function realised by S_1 . Since all descendants of v were gates from Ω_1 it follows that $f_1 \leq_\alpha f$. S_1 contains at most $q-1$ Ω_2 gates and this complete the proof by induction of (a).

c) Let $f, g \in [\Omega_1 \cup \Omega_2]$, with $g \leq_\alpha f$, S_f be any $\Omega_1 \cup \Omega_2 \cup \{\alpha\}$ network computing f and S_g any $\Omega_1 \cup \Omega_2$ network realising the function g . Replace any occurrence of the constant function α in S_f by the output of the network S_g , i.e the function g . Let the resulting network be denoted S_1 and f_1 the function which it computes. We claim that $f = \theta_\alpha(f_1, g)$. It is sufficient to consider the case $\alpha = 1$

only, $\alpha = 0$ following by a similar argument. To see that $f = f_1 \wedge g$ consider the two possible values which g can assume on any assignment π . If $g(\pi) = 1$, then S_1 behaves exactly as S_f hence $f(\pi) = f_1(\pi)$. On the other hand suppose that $g(\pi) = 0$. Then by the choice of g and definition of \leq_α , we have $f \leq g$, hence $f(\pi) = 0$. The upper bound asserted by (c) is now immediate. \square

Corollary 3.17: Let $\Omega_1 = \{\wedge, \vee\} \subset M_2$ and define $\Omega_\alpha \subset B_2$, for $\alpha \in \{0, 1\}$ to be any subset of,

$$\begin{aligned} & \{\ominus, \oplus, \boxplus\} \text{ for } \alpha = 0 \\ & \{\oplus, \neg\ominus, \neg\boxplus\} \text{ for } \alpha = 1 \end{aligned}$$

$\langle \Omega_1, \Omega_\alpha, \alpha \rangle$ is sympathetic and for all $f \in M_n$ such that $\mathbf{C}_{\Omega_1}(f) = O(\mathbf{C}_{\Omega_1 \cup \Omega_\alpha}(f))$ it holds that,

$\forall g \in [\Omega_1 \cup \Omega_\alpha]$ with $g \leq f$

$$\mathbf{C}_{\Omega_1 \cup \Omega_\alpha \cup \{\alpha\}}(f) = \Omega(\mathbf{C}_{\Omega_1}(f) - \mathbf{C}_{\Omega_1 \cup \Omega_\alpha}(g)) \quad \square$$

The basis $\{\mu, \oplus\}$ has some interesting properties with respect to the question of realising the disjunction of 2 functions over disjoint sets of variables. As we noted earlier, for complete bases, there exist pairs of functions, f and g , for which $\mathbf{C}(f \vee g) \ll \mathbf{C}(f) + \mathbf{C}(g)$. No explicit examples of such behaviour are known and whether such savings are possible for monotone networks remains an open question. For the basis $\{\mu, \oplus\}$ we can exhibit a superpolynomial reduction in complexity.

Theorem 3.29: Let $f \in M_n$ with arguments \mathbf{X}_n and let y, z, u be Boolean variables not contained in \mathbf{X}_n . Define the function $F(\mathbf{X}_n, y, z)$ by

$$F(\mathbf{X}_n, y, z) = y(z \vee f) \vee (z \tilde{f})$$

and the function $G_f(\mathbf{X}_n, y, z, u)$ to be,

$$G_f = u \vee F(\mathbf{X}_n, y, z)$$

i) If

$$\mathbf{C}_{\{\mu, \varnothing\}}(F) \leq r \mathbf{C}_{\{\mu, \varnothing\}}(G_f)$$

for some function $r: \mathbf{N} \rightarrow \mathbf{N}$, then $\mathbf{C}(f) = \Omega(\mathbf{C}^m(f)/r(n))$.

ii) There is a function $f \in M_n$ for which $\mathbf{C}_{\{\mu, \varnothing\}}(F)$ is superpolynomial, but $\mathbf{C}_{\{\mu, \varnothing\}}(G_f) = O(n^{2.5})$.

Proof: (i) is easily derived using (3.55) and (3.56) (with $g=u$) together with the fact that G_f is easily computable given f . For (ii) the construction is explicit. Let $n=m^2$ and $f \in M_n$ be the Perfect Matching function, PM , considered in Section(3.5). From (3.55) and Theorem(3.18) we have

$$\mathbf{C}_{\{\mu, \varnothing\}}(F) \geq n^{c \log n}$$

for some $c > 0$. But from (3.56) and the fact that f has combinational complexity $O(n^{2.5})$, cf Corollary(3.11), it follows that

$$\mathbf{C}_{\{\mu, \varnothing\}}(G_f) = O(\mathbf{C}(G_f)) = O(n^{2.5})$$

So we have an explicitly defined function, $F \in B_{n+2}$ with arguments $\langle \mathbf{X}_n, y, z \rangle$ which has superpolynomial complexity over the basis $\{\mu, \varnothing\}$ but such that the function $u \vee F$ has polynomial complexity over the same basis. \square

Bibliographic Notes

Alekseev (1973) considers the more general problem of counting the number of k -valued monotone functions. Beynon (1985) presents

an algebraic interpretation of replacement rules. McColl (1978a) proves an upper bound of $n + 1$ on the depth of monotone networks realising functions in M_n .

Apart from those presented above there are a number of results on the complexity of sets of monotone functions. Efficient constructions of sorting networks are given in (Ajtai et al., 1984) and improvements to this by Paterson (1987). Lower bounds on various functions may be found in Van Voorhis (1972); Lamagna (1979); Lamagna and Savage (1974); Pippenger and Valiant (1976); and Tarjan (1978). The lower bound on monotone matrix product from Paterson (1975) and Mehlhorn and Galil (1976) improves an earlier result of Pratt (1975).

For single output functions, Long (1986) gives a complicated proof that $C^m(MAJ_n) \geq 4n$. The results of Razborov concerning clique functions can be used to obtain lower bounds on the monotone complexity of *SAT* and Hamiltonian cycle; Skyum and Valiant (1985) discusses monotone projections between these families. Jukna (1986) and Andreev (1987) outline alternative methods of deriving exponential lower bounds. A conjecture of Schnorr (1976c) which would have yielded similar bounds has since been refuted in Wegener (1979). Razborov (1988a) examines the possibility of extending the approximation method to networks over complete bases.

A number of papers consider the power of negation in various senses. Negation limited networks being examined in Fischer (1974) and Markov (1957). Skyum (1983) considers an interpretation in which negation is exponentially powerful for computing Boolean functions. Valiant (1979b) has shown that negation is exponentially powerful for computing arithmetic functions. Dunne (1985c) gives a more general characterisation theorem for pseudo-complements which based on ideas first used by Wegener (1986).

Alon and Boppana (1986) also prove results on the relative numbers of \wedge and \vee gates required to compute monotone functions. Specifically they show that if $f \in M_n$ can be computed by a monotone network containing $k \geq 1$ \wedge -gates, then f can be computed by a monotone network, S , with k \wedge -gates and $\mathbf{C}^m(S) \leq kn + \binom{k-1}{2} - 1$.

Galibati and Fischer (1981) consider realising pairs of monotone functions on disjoint sets of variables, proving that $\mathbf{C}^m(\{f, g\}) = \mathbf{C}^m(f) + \mathbf{C}^m(g)$. Lenz and Wegener (1987) examine the number of \wedge -gates required to compute functions in $\mathcal{Q}_{n,2}$, developing the work of Mirwald and Schnorr (1987) on the conjunctive complexity of quadratic forms, i.e ringsum expansions in which every product is of length exactly 2. Finally Rivest (1977) has shown that monotone sequential machines (i.e monotone circuits with feedback loops) may be more efficient than monotone networks.

Chapter 4

Formulae

*But let your communication be, Yea, Yea; Nay, Nay:
for whatsoever is more than these cometh of evil.* **Matt. v. 32**

The complexity theory of realising Boolean functions by Ω -formulae, as introduced in Defn(1.3), has its roots in the study of relay-contact networks and their properties. The mathematical investigation of relay-contact schemes was originated, independently, by Shannon (1938) in the U.S, Shestakov (1938) in the Soviet Union, and Nakasima (1936) in Japan. So this restricted model has a history as old as, and to some degree independent of, the theory of combinational complexity. Although the theory of relay-contact circuits is no longer technologically relevant, the facts that combinational network depth and formula depth are equivalent; that lower bounds on formula depth may be deduced from similar bounds on formula size, cf Theorem(2.4); and that Ω -formulae are a model to which a considerable literature has been devoted, justify a substantial treatment of it. The aim of the present chapter is to offer such a presentation.

In Section(4.1) the lower bound of Riordan and Shannon (1942) on the formula size of almost all Boolean functions is given. This is a counting argument similar in spirit to Theorem(2.6). In the same section the asymptotically matching upper bound from Lupanov (1962) is derived. The (k, s) -Lupanov decomposition, the description of which preceded Theorem(2.7), is employed to the same effect as in the analogous upper bound on combinational complexity.

Progress in obtaining non-trivial lower bounds on formula size has been considerably more advanced than with combinational

complexity. A number of general techniques, applicable to formulae over the basis B_2 have been established. Section(4.2) describes the most important of these: the technique of Neciporuk (1966) and further applications of this from Harper and Savage (1972), and Schürfeld (1983); the methods of Hodes and Specker (1968) as enhanced by Pudlak (1983); and the approach of (Fischer et al., 1982).

Section(4.3) considers further the relation between formula size and depth. The main results presented are size/depth trade-offs obtained by Commentz-Walter (1979) and (Commentz-Walter and Sattler, 1980).

In Section(4.4) we consider some efficient constructions of formulae for specific Boolean functions, in particular for symmetric Boolean functions.

Section(4.5) concludes this chapter and looks at formulae over bases other than B_2 . The techniques of Khrapchenko (1971a, b) and Andreev (1986) are described. These yield lower bounds on formulae over the basis $\{\wedge, \vee, \neg\}$.

4.1) Bounds on Formula Size for almost all Boolean functions

Formulae restrict gates to having fanout at most 1, thus in graph-theoretic terms, the networks may be viewed as trees in which internal nodes are labelled with gate operations and the leaves by literals and constants; usually several leaves are labelled with the same literal. Let $\bar{\mathbf{X}}_n$ denote the set of literals $\{\bar{x}_1, \dots, \bar{x}_n\}$. Regarding formulae as trees allows us to associate a word of length $2\mathbf{L}(F) + 1$ over the alphabet $\mathbf{X}_n \cup \bar{\mathbf{X}}_n \cup B_2$ with any formula F as follows,

Definition 4.1: $PREFIX : Formulae \rightarrow \{\mathbf{X}_n, \bar{\mathbf{X}}_n, B_2\}^*$ is the (injective) mapping from formulae, F , onto words of length $2\mathbf{L}(F) + 1$, defined

inductively by:

P1) If $F = x_i$ or $F = \bar{x}_i$ then $PREFIX(F) = F$

P2) If $F = F_1 * F_2$, where $*$ $\in B_2$ then,

$$PREFIX(F) = * PREFIX(F_1) PREFIX(F_2) \quad \bullet$$

Now as a consequence of Lemma(1.3) every gate in an optimal formula over the basis B_2 depends on both its inputs. Thus minimal formulae do not contain projections or constant functions as gate operations.

Theorem 4.1: (Riordan and Shannon, 1942) For all $\epsilon > 0$ and n sufficiently large. For almost all $f \in B_n$,

$$L(f) > \frac{(1 - \epsilon)2^n}{\log n}$$

Proof: We estimate the number of distinct optimal formulae over basis B_2 which contain at most M gates. As in the proof of Theorem(2.6), we can show that if $M \leq \frac{(1 - \epsilon)2^n}{\log n}$ for any $\epsilon > 0$ then the number of these is $o(|B_n|)$.

Let $L(M)$ denote the number of distinct minimal formulae containing at most M gates and $l(m)$ the number with *exactly* m gates.

Obviously $L(M) = \sum_{m=0}^M l(m)$ so we need only bound the quantity $l(m)$.

With Defn(4.1) we can associate a unique word in $\{\mathbf{X}_n, \bar{\mathbf{X}}_n, B_2\}^{2m+1}$ with any formula of size m by using the mapping $PREFIX$. Now for any formula F of size m , $PREFIX(F)$ contains exactly m operator symbols, i.e from B_2 , and exactly $m + 1$ literal symbols. There are at

most $\binom{2m+1}{m}$ choices for the positions of operator symbols, 10

choices for each operation and $2n$ choices for each literal. Therefore,

$$l(m) \leq \binom{2m+1}{m} 10^m (2n)^{m+1} \leq \sqrt{\frac{2}{\pi}} \frac{2^{3m+2} 10^m n^{m+1}}{\sqrt{2m+1}}$$

Hence,

$$L(M) = \sum_{m=0}^M l(m) \leq \sqrt{\frac{2}{\pi}} \sum_{m=0}^M \frac{2^{3m+2} 10^m n^{m+1}}{\sqrt{2m+1}}$$

which is

$$\leq \sqrt{\frac{2}{\pi}} \frac{2^{3M+2} 10^M n^{M+1}}{\sqrt{2M+1}} \leq 2^{\delta M} n^{M+1}$$

where δ is some constant. It is now easy to verify that if $M \leq \frac{(1-\varepsilon)2^n}{\log n}$ then

$$L(M) \leq 2^{(1-\varepsilon)2^n + \frac{\delta(1-\varepsilon)2^n}{\log n} + \log n}$$

which is $o(|B_n|)$ as required. \square

Lupanov (1962) gives a construction which asymptotically matches this lower bound. It is based on the following result of Finikov (1957) which proves an upper bound on the formula size of Boolean functions over \mathbf{X}_n with exactly r satisfying assignments. Obviously any such function, f , has formula size at most $r + r - 1 = 2r - 1$. Finikov's result improves this for "small" r .

Lemma 4.1: (Finikov, 1957) Let $f \in B_n$ such that f has exactly r satisfying assignments. Then

$$\mathbf{L}(f) \leq 2n - 1 + r2^{r-1}$$

Proof: Let $\{\alpha^{(1)}, \dots, \alpha^{(r)}\} \subseteq \{0, 1\}^n$ be the set of assignments such

that $f(\alpha^{(i)}) = 1$ for each $1 \leq i \leq r$. We may construct a table with r rows and n columns, as below

x_1	x_2	\dots	x_i	\dots	x_n
$\alpha_1^{(1)}$	$\alpha_2^{(1)}$	\dots	$\alpha_i^{(1)}$	\dots	$\alpha_n^{(1)}$
$\alpha_1^{(2)}$	$\alpha_2^{(2)}$	\dots	$\alpha_i^{(2)}$	\dots	$\alpha_n^{(2)}$
.....					
$\alpha_1^{(j)}$	$\alpha_2^{(j)}$	\dots	$\alpha_i^{(j)}$	\dots	$\alpha_n^{(j)}$
.....					
$\alpha_1^{(r)}$	$\alpha_2^{(r)}$	\dots	$\alpha_i^{(r)}$	\dots	$\alpha_n^{(r)}$

In this table $\alpha_i^{(j)}$ is the value given to x_i by $\alpha^{(j)}$. Now for any $\beta = \langle b_1, \dots, b_r \rangle \in \{0, 1\}^r$ let $MATCH_\beta$ be the subset $\{i_1, \dots, i_p\}$ of $\{1, 2, \dots, n\}$ such that $\forall i_j \in MATCH_\beta$

$$\alpha_{i_j}^{(k)} = b_k \text{ for each } 1 \leq k \leq r$$

i.e those columns which equal the r -tuple β when transposed into a row vector. Clearly if $\beta \neq \gamma$ then the set $MATCH_\beta \cap MATCH_\gamma$ is empty. With t denoting

$$|\{ \beta \in \{0, 1\}^r : MATCH_\beta \neq \emptyset \}|$$

$\langle \beta_1, \beta_2, \dots, \beta_t \rangle$ will be some ordering of those r -tuples, β , for which $MATCH_\beta$ is non-empty. To avoid an excess of subscripts $MATCH_i$ will be used instead of $MATCH_{\beta_i}$. We further simplify the notation by assuming that $MATCH_i$ consists of a contiguous sequence of indices, thus $MATCH_1 = \{1, 2, \dots, q_1\}$, $MATCH_i = \{q_{i-1} + 1, \dots, q_i\}$. This could always be arranged by renaming variables. It should be clear that, $\sum_{i=1}^t |MATCH_i| = n$.

We can now define two Boolean functions over \mathbf{X}_n , using the table above and the partition of its columns induced by the β_i . Conventionally $q_0 = 0$ below.

$$\begin{aligned} \text{row-traverse}(\mathbf{X}_n) &= \bigwedge_{i=1}^t \left(\bigwedge_{j=q_{i-1}+1}^{q_i} x_j \vee \bigwedge_{j=q_{i-1}+1}^{q_i} \bar{x}_j \right) \\ \text{last-col}(\mathbf{X}_n) &= \bigvee_{i=1}^r \bigwedge_{j=1}^t x_{q_j}^{\alpha_j^{(i)}} \end{aligned}$$

We now have that

$$f(\mathbf{X}_n) = \text{row-traverse}(\mathbf{X}_n) \text{last-col}(\mathbf{X}_n)$$

since

$$\begin{aligned} \text{row-traverse last-col} &= \bigwedge_{i=1}^t \left(\bigwedge_{j=q_{i-1}+1}^{q_i} x_j \vee \bigwedge_{j=q_{i-1}+1}^{q_i} \bar{x}_j \right) \left(\bigvee_{i=1}^r \bigwedge_{j=1}^t x_{q_j}^{\alpha_j^{(i)}} \right) \\ &= \bigvee_{i=1}^r \left(\bigwedge_{j=1}^t x_{q_j}^{\alpha_j^{(i)}} \bigwedge_{k=1}^t \left(\bigwedge_{l=q_{i-1}+1}^{q_i} x_l \vee \bigwedge_{l=q_{i-1}+1}^{q_i} \bar{x}_l \right) \right) \\ &= \bigvee_{i=1}^r \bigwedge_{k=1}^t \bigwedge_{l=q_{k-1}+1}^{q_k} x_l^{\alpha_l^{(i)}} \\ &= \bigvee_{i=1}^r \bigwedge_{k=1}^n x_k^{\alpha_k^{(i)}} = f(\mathbf{X}_n) \end{aligned}$$

Hence $\mathbf{L}(f) \leq \mathbf{L}(\text{row-traverse}) + \mathbf{L}(\text{last-col}) + 1$. From their definition,

$$\begin{aligned} \mathbf{L}(\text{row-traverse}) &\leq t - 1 + 2 \sum_{i=1}^t (|\text{MATCH}_i| - 1) + t \\ &= 2t - 1 + 2(n - t) = 2n - 1 \end{aligned}$$

$$\begin{aligned} \mathbf{L}(\text{last} - \text{col}) &\leq r - 1 + \sum_{i=1}^r (t - 1) \\ &= rt - 1 \end{aligned}$$

By negating variables if necessary, we can always guarantee that $\langle 1, 1, \dots, 1 \rangle$ satisfies f and so $t \leq 2^{r-1}$. This proves the lemma. \square

Finikov's result can also be used to improve the bound $rn - 1$ for "large" r .

Corollary 4.1: Let $f \in B_n$ be as in Lemma(4.1) with $r \geq (\log n)^2$. Then

$$\mathbf{L}(f) \leq \frac{2nr}{\log n} (1 + \xi(n))$$

where ξ is such that $\lim_{n \rightarrow \infty} \xi(n) = 0$.

Proof: Let $\{\alpha^{(1)}, \dots, \alpha^{(r)}\}$ be the satisfying assignments of f and partition these into q blocks, B_1, \dots, B_q where $|B_i| = d$ for $1 \leq i < q$ and $|B_q| \leq d$ for some d to be fixed subsequently. So $q \leq \lfloor r/d \rfloor + 1$. Clearly $f(\mathbf{X}_n) = \bigvee_{i=1}^q g_i(\mathbf{X}_n)$, where g_i is satisfied by exactly the assignments in B_i . Applying Lemma(4.1),

$$\mathbf{L}(f) \leq (2n - 1 + d 2^{d-1})q + q = (2n + d 2^{d-1})q$$

Fix $d = \log n - 2 \log \log n$. We then have,

$$\mathbf{L}(f) \leq \frac{2nr}{\log n - 2 \log \log n} + 2n + \frac{rn(\log n - 2 \log \log n)}{(\log n)^2}$$

and this is no more than

$$\frac{2nr}{\log n} \left(1 + \frac{c \log n}{r} + \frac{\log n - 2 \log \log n}{(\log n)^2} \right)$$

as required. \square

With these two results we can now derive an upper bound on $\mathbf{L}(f)$ for any $f \in B_n$.

Theorem 4.2: (Lupanov, 1962) $\forall f \in B_n, \forall \varepsilon > 0$ and n sufficiently large,

$$\mathbf{L}(f) < \frac{(1 + \varepsilon)2^n}{\log n}$$

Proof: Consider the (k, s) -Lupanov decomposition of f , as described above.

$$f(\mathbf{X}_n) = \bigvee_{i=1}^d \bigvee_{\mathbf{v} \neq \mathbf{0}} [\text{row} - \text{match}_{i,\mathbf{v}}(\mathbf{Y}) \wedge \text{col} - \text{match}_{i,\mathbf{v}}(\mathbf{Z})]$$

where

$$\text{row} - \text{match}_{i,\mathbf{v}}(\mathbf{Y}) = \bigvee_{\alpha \in \{0,1\}^k \cap R_i} \delta_\alpha(\mathbf{Y}) \wedge v(\alpha)$$

$$\text{col} - \text{match}_{i,\mathbf{v}}(\mathbf{Z}) = \bigvee_{\beta \in \{0,1\}^{n-k} \cap P_{i,\mathbf{v}}} \delta_\beta(\mathbf{Z})$$

So $\mathbf{L}(f)$ is at most

$$\sum_{i=1}^d \sum_{\mathbf{v} \neq \mathbf{0}} (\mathbf{L}(\text{row} - \text{match}_{i,\mathbf{v}}) + \mathbf{L}(\text{col} - \text{match}_{i,\mathbf{v}})) + 2d(2^s - 1) - 1$$

From the definition of $\text{row} - \text{match}_{i,\mathbf{v}}$ it is immediate that,

$$\sum_{i=1}^d \sum_{\mathbf{v} \neq \mathbf{0}} \mathbf{L}(\text{row} - \text{match}_{i,\mathbf{v}}) \leq d(2^s - 1)(ks - 1)$$

For an upper bound on $\text{col} - \text{match}_{i,\mathbf{v}}$ we proceed as follows. Let

$$C_i(t) = |\{ \mathbf{v} \in \{0,1\}^s : |P_{i,\mathbf{v}}| = t \}|$$

and $Q(r, m)$ be the maximal formula size of any function in B_m with exactly r satisfying assignments. Then $\sum_t C_i(t) \leq 2^s$ since the sets,

$$\{\{\mathbf{v} \in \{0, 1\}^s : |P_{i,\mathbf{v}}| = t\}\}_{1 \leq i \leq t}$$

are pairwise disjoint. Also $\sum_t tC_i(t) = 2^{n-k}$ since

$$\bigcup_{\mathbf{v} \in \{0,1\}^s} P_{i,\mathbf{v}} = \{0, 1\}^{n-k}$$

Now, for any i ,

$$\begin{aligned} \bigvee_{\mathbf{v} \neq \mathbf{0}} \text{col-match}_{i,\mathbf{v}} &= \bigvee_{\mathbf{v} \neq \mathbf{0}} \bigvee_{\beta \in \{0,1\}^{n-k} \cap P_{i,\mathbf{v}}} \delta_\beta(\mathbf{Z}) \\ &= \bigvee_t \bigvee_{\{\mathbf{v} \neq \mathbf{0} : |P_{i,\mathbf{v}}|=t\}} \bigvee_{\beta \in P_{i,\mathbf{v}} \cap \{0,1\}^{n-k}} \delta_\beta(\mathbf{Z}) \end{aligned}$$

Hence,

$$\sum_{i=1}^d \sum_{\mathbf{v} \neq \mathbf{0}} \mathbf{L}(\text{col-match}_{i,\mathbf{v}}) \leq \sum_{i=1}^d \sum_t Q(t, n-k) C_i(t)$$

This is,

$$\begin{aligned} &\leq \sum_{i=1}^d \sum_{t < (\log(n-k))^2} Q(t, n-k) C_i(t) + \sum_{i=1}^d \sum_{t \geq (\log(n-k))^2} Q(t, n-k) C_i(t) \\ &\leq \sum_{i=1}^d \sum_{t < (\log(n-k))^2} (tn - tk - 1) C_i(t) + \\ &\quad \sum_{i=1}^d \sum_{t \geq (\log(n-k))^2} \frac{2t(n-k)}{\log(n-k)} (1 + \xi(n-k)) C_i(t) \end{aligned}$$

$$\leq d2^s(n-k)(\log(n-k))^2 + \frac{2d(n-k)2^{n-k}}{\log(n-k)}(1 + \xi(n-k))$$

and so, $\mathbf{L}(f)$ does not exceed

$$d2^s(ks + (n-k)(\log(n-k))^2 + 2) + \frac{2d(n-k)2^{n-k}}{\log(n-k)}(1 + \xi(n-k)) + d - 1$$

Setting $k = \lceil 2 \log n \rceil$, $s = \lceil n - 3 \log n \rceil$ and recalling that $d \leq 2^k/s + 1$ proves the theorem. \square

4.2) General Lower Bound Techniques

Formulae over the basis B_2 form the first widely studied restricted model for which non-trivial lower bound on complexity were obtained. In this section three important techniques for deriving such bounds will be discussed. All can be applied to a broad variety of functions.

The first method presented is that of Neciporuk (1966) which relates the formula size of $f(\mathbf{X}_n)$ to the number of distinct subfunctions over $\mathbf{Y} \subseteq \mathbf{X}_n$ arising from partial assignments to $\mathbf{X}_n - \mathbf{Y}$. These methods can yield bounds of at best $\Omega(n^2/\log n)$; this optimum being attained in Neciporuk (1966) and remaining, to date, the best lower bound on formula size. Neciporuk's approach has also been applied to yield lower bounds by a number of authors for a considerably diverse range of problems. We describe its employment by Harper and Savage (1972), to the "Marriage Problem" and by Schürfeld (1983) to various graph-theoretic problems.

The other techniques; Hodes and Specker (1968), Pudlak (1983); (Fischer, Meyer, Paterson, 1982); are similar in style. Both deduce lower bounds on formula size via (different) theorems of the following form:

If $\mathbf{L}(f)$ is "small" then $f(\mathbf{X}_n)$ possesses Property-X
or equivalently, and more directly applicable to lower bound proofs,

If $f(\mathbf{X}_n)$ does not have Property-X then $\mathbf{L}(f) = \Omega(g(n))$.

The exact definition of Property-X differs for each technique. Hodes and Specker (1968) has been shown capable of yielding bounds of $\Omega(n \log^* n)$, cf Vilfan (1976). Pudlak (1983) improved the basic methods of Hodes and Specker to achieve lower bounds of $\Omega(n \log \log n)$. Since a lengthy case analysis would be required we do not give a complete exposition of this approach. (Fischer et al., 1982) deduce bounds of $\Omega(n \log n)$ on formula size.

It should be noted that all of these approaches have differing realms of application. Neciporuk (1966) and its successors yield moderate bounds on functions to which the others are not of significant value. However Neciporuk's method is of little use for establishing results on symmetric functions. Within this class (Fischer et al., 1982) obtain $\Omega(n \log n)$ bounds; Pudlak (1983) deriving $\Omega(n \log \log n)$ bounds for certain symmetric functions outwith the power of both techniques.

4.2.1) The Neciporuk Bound

Definition 4.2: Let $f(\mathbf{X}_n) \in B_n$ and $\mathbf{Y} = \{y_1, \dots, y_m\} \subset \mathbf{X}_n$.

$$N_f(\mathbf{Y}) = \left| \bigcup_{\sigma \in \{0,1\}^{n-m}} \{f^{\mathbf{X}_n - \mathbf{Y} := \sigma}(\mathbf{Y})\} \right|$$

Thus $N_f(\mathbf{Y})$ is the number of distinct subfunctions obtainable from f by setting $\mathbf{X}_n - \mathbf{Y}$ to constants. •

Theorem 4.3: (Neciporuk, 1966)

$\forall f \in B_n, \forall \mathbf{Y} = \{y_1, \dots, y_m\} \subset \mathbf{X}_n:$

$$\mathbf{L}(f) \geq \log_5 N_f(\mathbf{Y}) + \max_{\sigma \in \{0,1\}^m} f^{\mathbf{Y}:=\sigma}(\mathbf{X}_n - \mathbf{Y})$$

Proof: The proof below is due to Paterson (pers. comm) and yields the best known multiplicative constant in the lower bound. Earlier methods give $\log_{16} N_f(\mathbf{Y}) = \log N_f(\mathbf{Y})/4$ instead of $\log_5 N_f = \log N_f/2.518$.

Let F be an optimal formula, over the basis B_2 , realising $f(\mathbf{X}_n)$, and $\mathbf{Y} = \{y_1, \dots, y_m\}$ be any proper subset of \mathbf{X}_n . For any $x \in \mathbf{X}_n$ define $occ(x, F)$ to be the total number of leaves (i.e inputs) of F labelled with x or \bar{x} . Similarly for $W \subseteq \mathbf{X}_n$, $occ(W, F) = \sum_{x \in W} occ(x, F)$. Now,

$$\mathbf{L}(f) = \mathbf{L}(F) = occ(\mathbf{Y}, F) + occ(\mathbf{X}_n - \mathbf{Y}, F) - 1$$

and since F can be amended to realise any subfunction over $\mathbf{X}_n - \mathbf{Y}$ of f , simply by setting \mathbf{Y} to the appropriate assignment, it is sufficient to show

$$occ(\mathbf{Y}, F) \geq \log_5 N_f(\mathbf{Y})$$

in order to prove the theorem.

Let $\mathbf{Z} = \{z_1, z_2, \dots, z_{n-m}\}$ denote the variables $\mathbf{X}_n - \mathbf{Y}$ and $S_f(\mathbf{Y})$ the set of functions,

$$S_f(\mathbf{Y}) = \{g(\mathbf{Y}) : g(\mathbf{Y}) = f^{\mathbf{Z}:=\sigma}(\mathbf{Y}), \sigma \in \{0, 1\}^{n-m}\}$$

Obviously $|S_f(\mathbf{Y})| = N_f(\mathbf{Y})$. Consider the relation \sim defined over $S_f(\mathbf{Y})$ by saying that $f \sim g$ if $f = g$ or $f = \neg g$. Clearly \sim is an equivalence relation and we denote by $M_f(\mathbf{Y})$ the number of equivalence classes of \sim , excluding any consisting of constant functions. With this definition,

$$M_f(\mathbf{Y}) \leq N_f(\mathbf{Y}) \leq 2 M_f(\mathbf{Y}) + 2$$

We claim that,

$$occ(\mathbf{Y}, F) \geq \log_5(4M_f + 1) > \log_5 N_f$$

The second inequality being immediate from the previous relation we proceed by induction on $\mathbf{L}(f) = \mathbf{L}(F) \geq 0$ to prove the first.

For the inductive base $\mathbf{L}(f) = 0$, f is just a single variable x . If $x \in \mathbf{Y}$ then $S_f(\mathbf{Y}) = \{x\}$ hence $M_f(\mathbf{Y}) = 1$. Thus,

$$occ(\mathbf{Y}, F) = 1 = \log_5 5 = \log_5(4M_f(\mathbf{Y}) + 1)$$

On the other hand if $x \notin \mathbf{Y}$ then $S_f(\mathbf{Y}) = \emptyset$ and so $M_f(\mathbf{Y}) = 0$. Thus,

$$occ(\mathbf{Y}, F) = 0 = \log_5 1 = \log_5(4M_f(\mathbf{Y}) + 1)$$

Assume the claim holds for all values $0 \leq \mathbf{L}(f) < t$. We show it holds for $\mathbf{L}(f) = t$ also. Let F be an optimal formula realising f of size $t \geq 1$. Then $F = G \theta H$ where G, H are formulae of size at most $t - 1$ realising functions $g(\mathbf{Y}, \mathbf{Z})$ and $h(\mathbf{Y}, \mathbf{Z})$ respectively. The inductive step follows from the fact that,

$$M_f(\mathbf{Y}) \leq 4M_g(\mathbf{Y})M_h(\mathbf{Y}) + M_g(\mathbf{Y}) + M_h(\mathbf{Y})$$

For with this we have,

$$\begin{aligned} \log_5(4M_f + 1) &\leq \log_5((4M_g + 1)(4M_h + 1)) \\ &= \log_5(4M_g + 1) + \log_5(4M_h + 1) \\ &\leq occ(\mathbf{Y}, G) + occ(\mathbf{Y}, H) = occ(\mathbf{Y}, F) \end{aligned}$$

by the inductive hypothesis.

To see that,

$$M_f \leq 4M_g M_h + M_g + M_h$$

note that as a consequence of Lemma(1.3)(ii), θ is either an \wedge -type

or an \oplus -type gate. Since M_f is certainly no more than the number of distinct functions obtainable by fixing occurrences of \mathbf{Z} in G and H independently we can produce an upper bound on M_f as follows.

Case 1: θ is \oplus -type

So $f = g \oplus h \oplus c$, where $c \in \{0, 1\}$. Any $f' \in S_f(\mathbf{Y})$ arises from one of the following cases,

$$\left\{ \begin{array}{l} 0 \oplus h', 1 \oplus h' : h' \in S_h \\ g' \oplus 0, g' \oplus 1 : g' \in S_g \\ g' \oplus h', 1 \oplus g' \oplus h' : g' \in S_g, h' \in S_h \end{array} \right\}$$

The first case contributes at most M_h classes to M_f ; the second at most M_g ; the last at most $M_g M_h$. Thus if θ is \oplus -type,

$$M_f \leq M_g M_h + M_g + M_h$$

Case 2: θ is \wedge -type.

So $f = (g^a \wedge h^b)^c$ where $a, b, c \in \{0, 1\}$. Here any $f' \in S_f(\mathbf{Y})$ arises from one of the cases below,

$$\left\{ \begin{array}{l} 1 \wedge h', 1 \wedge \neg h' : h' \in S_h \\ g' \wedge 1, \neg g' \wedge 1 : g' \in S_g \\ g' \wedge h', \neg(g' \wedge h') : g' \in S_g, h' \in S_h \\ g' \wedge \neg h', \neg(g' \wedge \neg h') : g' \in S_g, h' \in S_h \\ \neg g' \wedge h', \neg(\neg g' \wedge h') : g' \in S_g, h' \in S_h \\ \neg g' \wedge \neg h', g' \vee h' : g' \in S_g, h' \in S_h \end{array} \right\}$$

Again the first two cases contribute at most $M_g + M_h$ classes to M_f ; each of the remaining four contribute at most $M_g M_h$ each. Thus if θ is \wedge -type then,

$$M_f \leq 4M_g M_h + M_g + M_h$$

and this proves the theorem. \square

The results below are immediate from Theorem(4.3)

Corollary 4.2: Let $\langle \mathbf{X}^{(1)}, \dots, \mathbf{X}^{(m)} \rangle$ be a partition of \mathbf{X}_n . Then $\forall f \in B_n$

$$\mathbf{L}(f) \geq \sum_{i=1}^m \log_5 N_f(\mathbf{X}^{(i)}) - 1$$

Proof: Let F be any optimal formula for f . Then

$$\mathbf{L}(f) = \mathbf{L}(F) = \sum_{i=1}^m \text{occ}(\mathbf{X}^{(i)}, F) - 1$$

From the proof of Theorem(4.3), we have that for each $\mathbf{X}^{(i)}$,

$$\text{occ}(\mathbf{X}^{(i)}, F) \geq \log_5 N_f(\mathbf{X}^{(i)})$$

hence

$$\mathbf{L}(f) = \sum_{i=1}^m \text{occ}(\mathbf{X}^{(i)}, F) - 1 \geq \sum_{i=1}^m \log_5 N_f(\mathbf{X}^{(i)}) - 1$$

as claimed. \square

Corollary 4.3: Let n be an exact multiple of m and \mathbf{X}_n be partitioned into $\frac{n}{m}$ sets of m variables each. Let $\mathbf{X}^{(i)}$ denote the i 'th set, then for any $f \in B_n$,

$$\mathbf{L}(f) \geq \frac{n}{m} \min_{1 \leq i \leq m} \{ \log_5 N_f(\mathbf{X}^{(i)}) \}$$

Proof: Obvious from Corollary(4.2). \square

We now consider some specific applications of Neciporuk's method. The function originally used in Neciporuk (1966) is defined

as follows.

Let $n \in \mathbb{N}$ and set $m = \lceil \log n \rceil + 2$. $\mathbf{X}_{n,m}$ is a $\lceil n/m \rceil \times m$ matrix of Boolean variables $\{x_{i,j} : 1 \leq i \leq \lceil n/m \rceil, 1 \leq j \leq m\}$. Let $[\sigma_{ij}]$ be a $\lceil n/m \rceil \times m$ matrix of pairwise distinct Boolean m -tuples, each m -tuple containing at least two 1's. Neciporuk's function, $N(\mathbf{X}_{n,m})$, is given by,

$$N(\mathbf{X}_{n,m}) = \bigoplus_{\substack{1 \leq i \leq \lceil n/m \rceil \\ 1 \leq j \leq m}} \bigoplus_{\substack{k=1 \\ k \neq i}}^{\lceil n/m \rceil} x_{i,j} \wedge \bigwedge_{\{l : \sigma_{i,j}(l) = 1\}} x_{k,l}$$

Here $\sigma_{i,j}(l)$ denotes the l 'th bit of the m -tuple $\sigma_{i,j}$.

Before deriving a lower bound on $\mathbf{L}(N(\mathbf{X}_{n,m}))$ we require one property of the ringsum expansion.

Fact 4.1: Let $P = \{p_1, p_2, \dots, p_r\}$ and $Q = \{q_1, q_2, \dots, q_s\}$ be different sets of monoms over \mathbf{X}_n . Then,

$$\bigoplus_{i=1}^r p_i(\mathbf{X}_n) \neq \bigoplus_{i=1}^s q_i(\mathbf{X}_n)$$

Proof: Suppose the contrary, then from the definition of \oplus we have,

$$\bigoplus_{i=1}^r p_i \oplus \bigoplus_{i=1}^s q_i = 0$$

This is equivalent to,

$$\bigoplus_{p \in P \cap Q} p \oplus \bigoplus_{q \in P \cap Q} q \oplus \bigoplus_{p \in P - Q} p \oplus \bigoplus_{q \in Q - P} q = 0$$

i.e

$$\bigoplus_{p \in P - Q} p \oplus \bigoplus_{q \in Q - P} q = 0$$

Obviously $P - Q$ and $Q - P$ are disjoint sets and since $P \neq Q$ we have that the set $P - Q \cup Q - P$ is non-empty. Let m be a minimal product in this set, i.e a monom such that no proper subset of $var(m)$ defines an element of $P - Q \cup Q - P$. Then under the assignment α which fixes exactly the variables of m to 1, m is the only product in $P - Q \cup Q - P$ which is 1 and hence the left-hand side of this expression is $1 \neq 0$. This contradiction proves the result. \square

Lemma 4.2: $\forall 1 \leq i \leq \lceil n/m \rceil N_N(\mathbf{X}^{(i)}) = 2^{(\lceil n/m \rceil - 1)m}$

Proof: It suffices to show that any two distinct assignments to $\mathbf{X}_{n,m} - \mathbf{X}^{(i)}$ yield different subfunctions of $N(\mathbf{X}_{n,m})$. Consider two different assignments $\alpha = \langle a_1, \dots, a_p \rangle$ and $\beta = \langle b_1, \dots, b_p \rangle$ to the variables $\mathbf{X}_{n,m} - \mathbf{X}^{(i)}$, p denoting $(\lceil n/m \rceil - 1)m$. Since α and β are different there is some variable $x_{r,s} \in \mathbf{X}_{n,m} - \mathbf{X}^{(i)}$ such that $x_{r,s}^\alpha \neq x_{r,s}^\beta$. Without loss of generality we assume that $x_{r,s} = 1$ under α and 0 under β . The function $N^\alpha(\mathbf{X}^{(i)})$ is the \oplus over some set of monoms depending on $\mathbf{X}^{(i)}$. In particular, since $N(\mathbf{X}_{n,m})$ contains the product

$$x_{r,s} \wedge \bigwedge_{\{l : \sigma_{r,s}(l)=1\}} x_{i,l}$$

this subfunction contains the product, of at least two variables,

$$\bigwedge_{\{l : \sigma_{r,s}(l)=1\}} x_{i,l}$$

which is not contained in $N^\beta(\mathbf{X}^{(i)})$. Thus from Fact(4.1) the subfunctions N^α and N^β are different and the lemma follows. \square

Theorem 4.4: $\mathbf{L}(N(\mathbf{X}_{n,m})) = \Omega\left(\frac{n^2}{\log n}\right)$

Proof: From Corollary(4.3) and Lemma(4.2) we have,

$$\begin{aligned} \mathbf{L}(N(\mathbf{X}_{n,m})) &\geq \frac{n \log_5(2^{\lceil n/m \rceil - 1} m)}{m} \\ &= \frac{n(\lceil n/m \rceil - 1)m}{m \log 5} \geq \frac{n^2 - nm}{m \log 5} \end{aligned}$$

Since $m = \lceil \log n \rceil + 2$ this proves the theorem. \square

Harper and Savage (1972) show how Neciporuk's methods can be applied to yield a lower bound on the formula size of a function closely related to the Perfect Matching problem of Chapter(3).

Definition 4.3: Let $\mathbf{X}_{n,n} = \{x_{ij} : 1 \leq i, j \leq n\}$ be a set of n^2 Boolean variables where n is even. Recall that $B(\mathbf{X}_{n,n})$ is a mapping from assignments, α , to $\mathbf{X}_{n,n}$ onto $2n$ -vertex bipartite graphs with vertex set $V \cup W$ and that a matching in $B(\mathbf{X}_{n,n})$ is a subset of the edges of B such that each vertex is the endpoint of at most one edge. A k -matching is a matching with exactly k edges. k -Match($\mathbf{X}_{n,n}$) is the Boolean function which is true if and only if $B(\mathbf{X}_{n,n})$ contains a k -matching. The *Stable Marriage Problem, SMP* is that of determining the cardinality of a maximal matching in a given bipartite graph. Finally $PSMP(\mathbf{X}_{n,n})$ is the Boolean function whose value is $SMP(B(\mathbf{X}_{n,n})) \pmod{2}$, i.e the parity of the number of edges in a maximal matching. •

Clearly,

$$PSMP(\mathbf{X}_{n,n}) = \bigvee_{i=0}^{n/2} (2i+1) - \text{Match}(\mathbf{X}_{n,n}) \wedge \overline{(2i+2) - \text{Match}(\mathbf{X}_{n,n})}$$

where $(n+j) - \text{Match}(\mathbf{X}_{n,n}) \equiv 0$ for $j > 0$.

Theorem 4.5: (Harper and Savage, 1972)

$$\mathbf{L}(PSMP(\mathbf{X}_{n,n})) = \Omega(N^{3/2})$$

where $N = n^2$.

Proof: Partition $\mathbf{X}_{n,n}$ into n blocks, P_k , where $0 \leq k \leq n-1$, defined by

$$P_k = \{ x_{ij} : i \equiv j + k \pmod{n} \}$$

Now since $SMP(B(\mathbf{X}_{n,n}))$ is invariant under relabelling of the graph vertices we have that $N_{PSMP}(P_0) = N_{PSMP}(P_i)$ for each $0 \leq i \leq n-1$. So it is sufficient to prove a large enough bound on $N_{PSMP}(P_0)$ and then appeal to Corollary(4.3). Consider the $2^{n^2/4}$ assignments to $\mathbf{X}_{n,n} - P_0$ in which $x_{ij} = 1 \Leftrightarrow i \leq n/2 < j$; thus $n/2 > i$ or $j \leq n/2 \Leftrightarrow x_{ij} = 0$. Let α and β be distinct assignments within this class. Then we can find r and s such that $r \leq n/2 \leq s$ and $x_{rs} = 0$ under α but $x_{rs} = 1$ under β . We claim that the functions $PSMP^\alpha(P_0)$ and $PSMP^\beta(P_0)$ are distinct. To see this note that $P_0 = \{x_{ii} : 1 \leq i \leq n\}$ and consider the bipartite graphs which result from the assignment γ to P_0 given by,

$$x_{ii} = 1 \iff i \neq r \text{ and } i \neq s$$

The graph $B(\alpha, \gamma)$ contains a matching of cardinality $n-2$ consisting of the edges $\{v_i, w_i\} : 1 \leq i \neq r, s \leq n\}$ but none of cardinality $n-1$ since the vertices v_r, v_s are unmarried. On the other hand the graph $B(\beta, \gamma)$ contains a matching of cardinality $n-1$ consisting of the same edges and $\{v_r, w_s\}$, but no matching of cardinality n since the vertex v_s is unmarried. Since $(n-2) \pmod{2} \neq (n-1) \pmod{2}$ the claim follows.

From the above argument $N_{PSMP}(P_0) \geq 2^{n^2/4}$. Hence from Corollary(4.3),

$$\mathbf{L}(PSMP(\mathbf{X}_{n,n})) \geq \frac{nn^2}{4 \log 5} = \Omega(N^{3/2})$$

as required. \square

Kloss (1966) employs the same partition of $\mathbf{X}_{n,n}$ to produce a lower bound on formula size for $\mathbf{GF}(2)$ determinant computation. Thus regarding $\mathbf{X}_{n,n}$ as an $n \times n$ matrix of Boolean variables,

$$DET(\mathbf{X}_{n,n}) = \bigoplus_{\sigma \in S_n} \bigwedge_{i=1}^n x_{i, \sigma(i)}$$

Theorem 4.6: (Kloss, 1966) $L(DET(\mathbf{X}_{n,n})) = \Omega(N^{3/2})$, where $N = n^2$.

Proof: Exercise. \square

Our final example using Neciporuk’s technique is taken from Schürfeld (1983) which derives superlinear lower bounds for the k -clique function, introduced in Chapter(3). Schürfeld (1983) improves the lower bounds of Mamatov (1979) for this function, which were also obtained using Neciporuk’s argument.

Theorem 4.7: (Schürfeld, 1983) Let \mathbf{X}_n^U be a set of $n/2$ Boolean variables encoding the possible edges of an n -vertex undirected graph as before. For all k , $3 \leq k \leq n$,

$$L(k - clique(\mathbf{X}_n^U)) = \Omega((n - k)^3)$$

Proof: We describe a partition of \mathbf{X}_n^U into $n - k + 3$ disjoint sets of edges, T_1, \dots, T_p and show that for each i , $2 \leq i \leq p$, $N_{k-clique}(T_i) \geq 2^{(n-k-i+3)^2/4}$. The partition classes are given as,

$$T_1 = \{ x_{ij} : 1 \leq i \leq k - 3, i + 1 \leq j \leq n \}$$

$$T_i = \{ x_{ij} : i + 1 \leq j \leq n \} \text{ for } 2 \leq i \leq n - k - 3 \equiv p$$

Consider any subset T_i , for some $i \geq 2$. We wish to identify a collection of $2^{(n-k-i+3)^2/4}$ assignments to $\mathbf{X}_n^U - T_i$ which give rise to distinct functions over T_i . Now any assignment α to $\mathbf{X}_n^U - T_i$ partially describes an n -vertex undirected graph, $G(\alpha)$, in which only the edges

associated with T_i variables are unspecified. So two distinct assignments, α and β , yield different functions on T_i if and only if $G(\alpha)$ can be extended to a graph containing a k -clique and $G(\beta)$ to one not containing any k -clique, by the same fixation of the unspecified edges T_i .

We claim that the following class of assignments to $\mathbf{X}_n^U - T_i$ has the desired property.

$$\left\{ \begin{array}{l} \left\{ \begin{array}{l} x_{ij} := 1 \text{ if } 1 \leq i \leq k-3, 2 \leq j \leq k-3 \\ x_{ij} := 1 \text{ if } 1 \leq i \leq k-3, k+i-4 \leq j \leq n \\ x_{ij} := 0 \text{ if } 1 \leq i \leq k-3, k-2 \leq j \leq k+i-5 \end{array} \right\} T_1 \text{ assignment} \\ \left\{ \begin{array}{l} x_{ij} := 0 \text{ if } k-2 \leq i \leq k+i-5, i+1 \leq j \leq n \end{array} \right\} T_h \text{ assignment } 2 \leq h \leq i-1 \\ \left\{ \begin{array}{l} \text{Any bipartite graph on vertices } k-i+3, \dots, n \end{array} \right\} T_h \text{ assignment } i+1 \leq h \leq p \end{array} \right\}$$

Distinct assignments in this class differ only in the choice of bipartite graph for the final part. Let $G(\alpha)$ be the partial graph defined by any assignment α in this set. First note that this graph contains several $k-1$ -cliques, consisting of the vertices $1, 2, \dots, k-3$ and any two of the vertices $k+i-3, k+i-2, \dots, n$. However $G(\alpha)$ does not contain a k -clique; suppose the contrary and that v_1, \dots, v_k were the vertices in a k -clique of $G(\alpha)$. If one of these vertices is $k-i+4$ then at least two, v and w say, must be in the set $\{k+i-3, \dots, n\}$. But the edges $\{k-i+4, v\}$ and $\{k-i+4, w\}$ are unspecified in $G(\alpha)$. The only other possibility is that at least 3 vertices, u, v and w say, from $\{k+i-3, \dots, n\}$ form part of the k -clique, and so $\{u, v, w\}$ is a 3-clique. But by the construction u, v and w are vertices in a bipartite graph, and it is well-known that any bipartite graph does not contain a

3-clique, or more generally any odd length cycle. These contradictions establish that $G(\alpha)$ does not contain a k -clique and hence the subfunction over T_i resulting by fixing $\mathbf{X}_n^U - T_i$ to α is not the constant function 1 (or the constant function 0 from the first part of the argument above). Finally we note that the only unspecified edges of $G(\alpha)$ are those between the vertex $k+i-4$ and the vertices, $\{k+i-3, \dots, n\}$ of the bipartite graph. All edges $\{j, k+i-4\}$ for $1 \leq j \leq k-3$ being present; all edges $\{j, k+i-4\}$ for $k-2 \leq j \leq k+i-5$ being absent.

Now let α and β be distinct assignments to $\mathbf{X}_n^U - T_i$ within this category. There is some edge x_{rs} for which $x_{rs}^\alpha \neq x_{rs}^\beta$. By the choice of α, β it must be the case that $k+i-3 \leq r \leq n-1, r+1 \leq s \leq n$. Without loss of generality assume that $\{r, s\}$ is an edge of $G(\alpha)$ but is absent from $G(\beta)$, i.e. $x_{rs}^\alpha = 1$. We extend the graphs $G(\alpha)$ and $G(\beta)$ with the following assignment to T_i ,

$$\left\{ \begin{array}{l} x_{k+i-4,r} := 1 \\ x_{k+i-4,s} := 1 \\ x_{k+i-4,j} := 0 \quad j \neq r, j \neq s, j \geq k+i-3 \end{array} \right\}$$

Let $H(\alpha), H(\beta)$ be the resulting completely specified graphs. $H(\alpha)$ has a k -clique consisting of the vertices

$$\{1, 2, \dots, k-3, k+i-4, r, s\}$$

However $H(\beta)$ contains no k -clique. For from the earlier argument on the non-existence of k -cliques in $G(\beta)$, any k -clique in $H(\beta)$ must use exactly 2 vertices from the set $\{k+i-3, \dots, n\}$ and the vertex $k+i-4$. From the construction of the assignment to T_i , $k+i-4$ is joined to only two vertices, r and s , in this set and these are not connected by an edge in $G(\beta)$ or $H(\beta)$.

We have thus established that different assignments in the class considered yield distinct functions of T_i .

It remains to establish the number of suitable assignments available. First note that the number of different bipartite graphs with q vertices is at least $2^{\lceil q/2 \rceil \lfloor q/2 \rfloor} > 2^{(q-1)^2/4}$. This is easily seen by considering the complete bipartite graph, $K_{\lceil q/2 \rceil, \lfloor q/2 \rfloor}$ and observing that the subgraphs defined by different subsets of its edges are distinct. This shows that $N_{k\text{-clique}}(T_i) \geq 2^{(n-k-i+3)^2/4}$. Applying Corollary(4.2) we have,

$$\begin{aligned} \mathbf{L}(k\text{-clique}) &\geq \sum_{i=2}^p \frac{\log N_{k\text{-clique}}(T_i)}{\log 5} \\ &\geq \frac{1}{\log 5} \sum_{i=2}^{n-k+3} \frac{(n-k-i+3)^2}{4} \\ &= \frac{1}{4 \log 5} \sum_{i=1}^{n-k+1} i^2 \\ &= \frac{1}{12 \log 5} (n-k+1) \left(n-k+\frac{3}{2}\right) (n-k+2) \\ &= \frac{1}{12 \log 5} (n-k)^3 - o((n-k)^3) \end{aligned}$$

which is the lower bound asserted in the theorem statement. \square

The bound of Thm(4.6) is $o(n)$ if $n-k = o(n^{2/3})$. For these cases Schürfeld (1983) derived the following bound which we state without proof.

Theorem 4.8: Let $h : \mathbf{N} \rightarrow \mathbf{N}$ be a function such that $h(n) \rightarrow \infty$ and $h(n) \leq \lfloor n/2 \rfloor$. Then

$$\mathbf{L}((n - h(n)) - \text{clique}) = \Omega(n^2 h(n)) \quad \square$$

4.2.2) The Hodes/Specker/Pudlak Lower Bound

Neciporuk's method turns out to be incapable of yielding non-trivial bounds on the formula size of symmetric Boolean functions. To see this consider any $f \in S_n$ and $\mathbf{Y} \subset \mathbf{X}_n$. Let α, β be different assignments to $\mathbf{X}_n - \mathbf{Y}$ containing the same number of 1's. Since f is symmetric it follows that $f^{|\alpha}(\mathbf{Y}) = f^{|\beta}(\mathbf{Y})$ and hence $N_f(\mathbf{Y}) \leq \min \{2^{|\mathbf{Y}|+1}, n - |\mathbf{Y}| + 1\}$. In the context of Corollaries(4.2) and (4.3) this establishes that at best linear lower bounds can be attained.

In this section we consider a bounding method discovered by Hodes and Specker (1968) with which superlinear lower bounds for symmetric functions can be proved. Our presentation follows that of Pudlak (1983); this paper improving the general theorem derived by Hodes and Specker. The basic idea underlying Hodes and Specker's original argument is quite simple, as is that behind Pudlak's enhancement; the complications in the proofs arise as a result of detailed case analyses, which are omitted in the description below.

Let $f \in B_n$. For $\mathbf{Y} \subseteq \mathbf{X}_n$, the \mathbf{Y} -restriction of $f(\mathbf{X}_n)$ is the subfunction of f obtained by fixing all variables in $\mathbf{X}_n - \mathbf{Y}$ to 0. The lower bound theorem of Hodes and Specker (1968) asserts that if $\mathbf{L}(f)$ is "small" then there is a \mathbf{Y} -restriction of f with a very precisely defined form. Specifically they proved,

Theorem 4.9: (Hodes & Specker, 1968) There exists a function, $\xi(r, n)$, such that for each r , $\lim_{n \rightarrow \infty} \xi(r, n) \rightarrow \infty$ and for which,

If $\mathbf{L}(f) \leq n \xi(r, n)$ then there exists $\mathbf{Y} \subset \mathbf{X}_n$, with $|\mathbf{Y}| = r$ and such that the \mathbf{Y} -restriction of f is equivalent to

$$\left(\bigoplus_{y \in \mathbf{Y}} y \right) \theta \left(\bigvee_{y \in \mathbf{Y}} y \right)$$

for some $\theta \in B_2$. \square

For r fixed, Vilfan (1976) established that $\xi(r, n)$ grew no faster than $\log^* n$. Pudlak (1983) used a different approach to show that the condition $\mathbf{L}(f) \leq n \xi(r, n)$ could be sharpened to,

$$\mathbf{L}(f) \leq \varepsilon n (\log \log n - \log r) \quad \forall \varepsilon > 0$$

It is this result with which we will be concerned in the remainder of this section. Some preliminary ideas are required.

Let $F(\mathbf{X}_n)$ and $G(\mathbf{Y}_n)$ be formulae having inputs labelled with variables from \mathbf{X}_n resp. $\mathbf{Y}_n = \{y_1, \dots, y_n\}$. We say that F and G are *isomorphic*, denoted $F \equiv_{iso} G$, if and only if

$$PREFIX(F(\mathbf{X}_n)) = PREFIX(G(y_1/x_1, \dots, y_n/x_n))$$

where $PREFIX$ is the mapping of Definition(4.1) and $G(y_1/x_1, \dots, y_n/x_n)$ denotes the formula G with each input x_i , (\bar{x}_i) replaced by the input y_i , (\bar{y}_i) . Obviously if $F \equiv_{iso} G$ then F and G represent the same n -input Boolean function (with different argument sets). However the converse is not necessarily true; two different formulae representing $f \in B_n$ may not be isomorphic.

If $F(\mathbf{X}_n)$ is a formula and $\mathbf{Y} \subseteq \mathbf{X}_n$, then the formula \mathbf{Y} -induced from F , denoted $F_{\mathbf{Y}}$, is that formula constructed from F as follows:

- I1) Replace all inputs $x \in \mathbf{X}_n - \mathbf{Y}$ by the constant 0.
- I2) Replace all subformulae $G_1 \theta G_2$ where both G_i are constant, by the appropriate constant value.
- I3) Replace all subformulae $G_1 \theta G_2$ where only one G_i is constant, $i = 1$ say, by $G_2, \neg G_2, 0$, or 1 as appropriate.

I4) Replace any subformula $\neg(\neg G)$ by G .

It should be clear that $occ(\mathbf{Y}, F_{\mathbf{Y}}) \leq occ(\mathbf{Y}, F)$. In addition if $\mathbf{Z} \subseteq \mathbf{Y}$ then $F_{\mathbf{Z}} \equiv_{iso} (F_{\mathbf{Y}})_{\mathbf{Z}}$.

Finally we define the concept of *homogeneity* which will be important in the subsequent argument. If $F(\mathbf{X}_n)$ is a formula and $\mathbf{Y} \subseteq \mathbf{X}_n$ we say that F is *homogeneous over \mathbf{Y}* if and only if $\forall \{y_{i_1}, y_{i_2}\} \subseteq \mathbf{Y}, \{y_{j_1}, y_{j_2}\} \subseteq \mathbf{Y}$ we have

$$F_{\{y_{i_1}, y_{i_2}\}} \equiv_{iso} F_{\{y_{j_1}, y_{j_2}\}}$$

In essence Pudlak's argument is as follows: if $F(\mathbf{X}_n)$ is homogeneous over \mathbf{Y} , with $|\mathbf{Y}| = r$ then the induced formula $F_{\mathbf{Y}}$ falls into one of 5 distinct classes; any formula in these classes is equivalent to some function of the form,

$$\left(\bigoplus_{y \in \mathbf{Y}} y \right) \theta \left(\bigvee_{y \in \mathbf{Y}} y \right)$$

Now if $F(\mathbf{X}_n)$ is a k -formula, i.e. $occ(x_i, F) \leq k$ for each $1 \leq i \leq n$, then any induced formula is again a k -formula and so we can bound from above the number of *non-isomorphic* k -formulae of 2 variables. By labelling each $\{x_i, x_j\} \subset \mathbf{X}_n$ with a distinct colour depending on the structure of $F_{\{x_i, x_j\}}$ we can identify, courtesy of Ramsey's theorem, a set $\mathbf{Z} \subseteq \mathbf{X}_n$ with the property that all the pairs in \mathbf{Z} have been identically coloured, i.e. F is homogeneous over \mathbf{Z} . The remainder of the proof derives an upper bound on size of k needed for the main theorem to work.

In this outline the first part is the most lengthy and its proof will be omitted.

Before presenting Pudlak's argument in greater detail we review the important combinatorial result alluded to in the description above.

With some slight notational abuse, we use V^r to denote all subsets of a set V , which contain exactly r elements. A k -colouring of a set V , is a mapping $\chi : V \rightarrow \{1, 2, \dots, k\}$. For a k -colouring χ of a set V , χ_j denotes the set

$$\{v \in V : \chi(v) = j\}$$

Definition 4.4: Let $V = \{v_1, \dots, v_m\}$ be a finite set, and $h \geq 1$, $l \geq 2$, $r_1, \dots, r_l \geq h$ integers. V has the $h - (r_1, \dots, r_l)$ -Ramsey property if and only if:

For all l -colourings, χ of V^h , there exists $W \subseteq V$ such that for some $k \in \{1, 2, \dots, l\}$, $W^h \subseteq \chi_k$ and $|W| = r_k$. •

Let $R_l^h(r_1, \dots, r_l)$ denote the smallest value m for which any set with m elements has the $h - (r_1, \dots, r_l)$ -Ramsey property (obviously the property is monotonic, so if it holds for an m -element set it holds also for an $(m + 1)$ -element set). The following is a classical result from combinatorial theory.

Fact 4.2: (Ramsey, 1930) For all $h \geq 1$, $l \geq 2$, $r_1, \dots, r_l \geq 2$, $R_l^h(r_1, \dots, r_l)$ exists. □

For our purposes an upper bound on this quantity, when $h = 2$, is required. The bound presented below is adapted from a proof of a simplified version of Fact(4.2) (for the case $l = 2$) given in Erdős and Spencer (1974).

Fact 4.3: Let $t = \sum_{j=1}^l r_j$. Then,

$$R_l^2(r_1, \dots, r_l) \leq l^t$$

Proof: By induction on $l \geq 2$. For the inductive base it may be shown that

$$R_2^2(p, q) \leq \binom{p+q-2}{p-1}$$

(see for example (Erdős and Spencer, 1974) pp.22-23, or Berge (1979), pp. 436-437).

$$\text{Hence } R_2^2(p, q) \leq 2^{p+q-2} < 2^{p+q}.$$

So inductively assume that the result holds for all values $2 \leq l' < l$. For the inductive step note that

$$R_l^2(r_1, \dots, r_l) = R_l^2(r_{\sigma(1)}, \dots, R_{\sigma(l)})$$

for any permutation σ , hence without loss of generality we may assume that $2 \leq r_1 \leq \dots \leq r_l$. With this convention a total lexicographic ordering may be defined over pairs of l -tuples in the obvious way. We prove the inductive step by a sub-induction over this ordering. First of all observe that $R_l^2(2, 2, \dots, 2) = 2$ and that $R_l^2(2, r_2, \dots, r_l) \leq R_{l-1}^2(r_2, \dots, r_l)$. In the former case the result clearly holds, in the latter we can appeal to the inductive hypothesis for l which now confirms $R_l^2(2, r_2, \dots, r_l)$ as at most

$$(l-1)^l < l^l$$

This establishes the subinductive base. For the step let $r_j \geq 3$ for each $1 \leq j \leq l$ and $V = \{v_1, \dots, v_m\}$ where $m = l^l$. We must show that V has the $2 - (r_1, \dots, r_l)$ -Ramsey property. Consider any l -colouring, χ , of V^2 . Choose any $v \in V$ and define

$$A_i = \{v_j : \{v, v_j\} \in \chi_i\}$$

Then $\sum_{i=1}^l |A_i| = m - 1$. Hence there exists some k , $1 \leq k \leq l$ for which,

$$|A_k| \geq \lceil (m-1)/l \rceil \geq l^l$$

By the subinductive hypothesis,

$$|A_k| \geq R_l^2(r_1, \dots, r_k - 1, \dots, r_l)$$

hence we can identify $W \subseteq A_k \subset V$ such that $W^2 \subseteq \mathcal{X}_m$ for some m and $|W| \geq r_m$. If $m \neq k$ then W is an appropriate subset of V ; if $m = k$, so that $|W| \geq r_k - 1$, then $W \cup \{v\}$ is an appropriate subset of V . This completes the proof of the upper bound. \square

We can now turn to the proof of the lower bound theorem.

Lemma 4.3: (Pudlak, 1983) Let $F(\mathbf{X}_n)$ be a formula which is homogeneous over \mathbf{X}_n , $n \geq 3$. Then F satisfies at least one of the following conditions.

H1) $F \equiv \text{constant}$

H2) $F(\mathbf{X}_n) = G(\mathbf{X}_n) \theta H(\mathbf{X}_n)$, for some $\theta \in B_2$ and both G and H are homogeneous over \mathbf{X}_n .

H3) $F(\mathbf{X}_n) = x_1 \theta (x_2 \theta (x_3 \cdots \theta (x_{n-1} \theta (x_n \phi G)) \cdots))$, or with the variable order reversed, where G is homogeneous over \mathbf{X}_n , and θ, ϕ satisfy

i) $y \phi z = y \theta z$ or $y \phi z = y \theta \bar{z}$ and

ii) $y \theta z$ is equivalent to one of $y \oplus z$, $y \vee z$, $\bar{y} \wedge z$ or z .

H4) $F(\mathbf{X}_n) = x_1 \theta (x_2 \theta (x_3 \cdots \theta (x_{n-1} \theta \phi(x_n)) \cdots))$, or with the variable order reversed, where one of the the following holds,

i) $y \theta z \equiv y \oplus z$ and $(\phi(z) = z \text{ or } \phi(z) = \bar{z})$.

ii) $y \theta z \equiv y \vee z$ and $\phi(z) = z$.

iii) $y \theta z \equiv \bar{y} \wedge z$ and $\phi(z) = \bar{z}$.

H5) $F(\mathbf{X}_n) = G[\phi(x_1), \dots, \phi(x_n)]$, where for some $\alpha \in \{0, 1\}$, one of the following holds:

- i) $G[y_1 , \dots , y_n] \equiv \alpha \oplus \bigoplus_{i=1}^n y_i.$
- ii) $G[y_1 , \dots , y_n] \equiv \alpha \oplus (\bigvee_{i=1}^n y_i)$ and $\phi(y) \equiv y.$
- iii) $G[y_1 , \dots , y_n] \equiv \alpha \oplus (\bigwedge_{i=1}^n y_i)$ and $\phi(y) \equiv \bar{y}.$

Proof: Omitted. \square

Lemma 4.4: If $F(\mathbf{X}_n)$ is homogeneous over \mathbf{X}_n , $n \geq 3$ then

$$F(\mathbf{X}_n) \equiv (\bigoplus_{i=1}^n x_i) \theta (\bigvee_{i=1}^n x_i)$$

for some $\theta \in B_2$.

Proof: The Lemma is easily established by induction on $\mathbf{L}(F(\mathbf{X}_n))$ from Lemma(4.3) using elementary Boolean operations. \square

Theorem 4.10: (Pudlak, 1983) Let $n \geq 3$. $\exists \varepsilon > 0$ such that $\forall f \in B_n$, $r \geq 3$ if $\mathbf{L}(f) \leq \varepsilon n (\log \log n - \log r)$ then for some $\mathbf{Y} \subseteq \mathbf{X}_n$ having $|\mathbf{Y}| = r$ it holds that,

$$f^{\mathbf{X}_n - \mathbf{Y} := \mathbf{0}}(\mathbf{Y}) \equiv (\bigoplus_{y \in \mathbf{Y}} y) \theta_f (\bigvee_{y \in \mathbf{Y}} y)$$

for some $\theta_f \in B_2$, depending on f .

Proof: Let $F(\mathbf{X}_n)$ be an optimal formula realising f and $P = \text{occ}(\mathbf{X}_n, F) = \mathbf{L}(f) + 1$. Define $\mathbf{Z} \subseteq \mathbf{X}_n$ as the set of variables,

$$\mathbf{Z} = \{ x_i \in \mathbf{X}_n : \text{occ}(x_i, F) \leq \lfloor 2P/n \rfloor \}$$

Then $|\mathbf{Z}| \geq n/2$ and the induced formula $G = F_{\mathbf{Z}}$ is a k -formula for which $k = \lfloor 2P/n \rfloor$. Any induced formula of G is also a k -formula. Using the method of Thm(4.1) it follows that the number of non-isomorphic k -formulae dependent on 2 variables is bounded above by $l = 2^{Ck}$ for some constant C . Thus using Fact(4.3), if

$$|\mathbf{Z}| \geq l^l \quad (4.1)$$

then we can find $\mathbf{Y} = \{y_1, \dots, y_r\} \subseteq \mathbf{Z} \subseteq \mathbf{X}_n$ such that all the formulae induced from G by pairs of variables from \mathbf{Y} are isomorphic, i.e. G is homogeneous over \mathbf{Y} , which implies that $G_{\mathbf{Y}}$ is homogeneous over \mathbf{Y} . Now applying Lemma(4.4) we have that the function realised by the induced formula $G_{\mathbf{Y}}$ is equivalent to

$$\left(\bigoplus_{y \in \mathbf{Y}} y \right) \theta_f \left(\bigvee_{y \in \mathbf{Y}} y \right)$$

From the definition of induced formula, the function computed by $G_{\mathbf{Y}}$ is just $f^{|\mathbf{X}_n - \mathbf{Y}| := \mathbf{0}}$. So we have identified a suitable restriction of f . It remains to relate this to $\mathbf{L}(f)$.

G is a k -formula. If,

$$k \leq \frac{\log \log |\mathbf{Z}| - \log r - \log C}{C + 1} \quad (4.2)$$

then,

$$Ck + \log k + \log C + \log r \leq (C + 1)k + \log C + \log r \leq \log \log |\mathbf{Z}|$$

hence, $2^{Ck} Ckr \leq \log |\mathbf{Z}| \Leftrightarrow l^l \leq |\mathbf{Z}|$, i.e. (4.2) \Leftrightarrow (4.1). Now $k = \lfloor 2P/n \rfloor$ and $|\mathbf{Z}| \geq n/2$. So from (4.2)

$$\frac{2\mathbf{L}(f) + 1}{n} \leq \frac{\log \log(n - 1) - \log r - \log C}{C + 1}$$

$$\mathbf{L}(f) \leq \frac{1}{2C + 1} \cdot n (\log \log(n - 1) - \log r - \log C) - 1/2$$

For $\varepsilon > 0$ sufficiently large, the right-hand side is no more than,

$$\varepsilon n (\log \log n - \log r)$$

for all $n \geq 3$, $r \geq 3$. \square

Corollary 4.4: Let $f \in B_n$ and $\mathbf{Y} \subseteq \mathbf{X}_n$ be a maximal cardinality subset of \mathbf{X}_n such that $f^{\mathbf{X}_n - \mathbf{Y} := \mathbf{0}}(\mathbf{Y}) \equiv (\bigoplus_{y \in \mathbf{Y}} y) \theta (\bigvee_{y \in \mathbf{Y}} y)$. (We say that $f(\mathbf{X}_n)$ has a $|\mathbf{Y}|$ -refinement in this case). Then,

$$\mathbf{L}(f) \geq \varepsilon n (\log \log n - \log(|\mathbf{Y}| + 1))$$

where ε is the constant of Thm(4.10).

Proof: If $\mathbf{L}(f) < \varepsilon n (\log \log n - \log(|\mathbf{Y}| + 1))$ then from Thm(4.10) we can find $\mathbf{Z} \subseteq \mathbf{X}_n$ such that $|\mathbf{Z}| \geq |\mathbf{Y}| + 1$ and

$$f^{\mathbf{X}_n - \mathbf{Z} := \mathbf{0}} \equiv (\bigoplus_{z \in \mathbf{Z}} z) \theta (\bigvee_{z \in \mathbf{Z}} z)$$

and this contradicts the choice of \mathbf{Y} . \square

As an application of Corollary(4.4) consider the following class of functions.

Definition 4.5: Let $f \in B_n$ and k be an integer in the range $1 \leq k \leq n - 3$. f is said to be k -sensitive if and only if $\forall \alpha = \langle a_1, \dots, a_n \rangle \in \{0, 1\}^n$

$$|\{i : a_i = 1\}| = k \iff f(\alpha) = 0, (1)$$

$$|\{i : a_i = 1\}| = k + 2 \iff f(\alpha) = 1, (0)$$

Theorem 4.11: If $f \in B_n$ is k -sensitive then $\mathbf{L}(f) \geq \delta \cdot n \log \log n$, for some $\delta > 0$.

Proof: Let f be k -sensitive and without loss of generality assume that $k \leq n/2$ (otherwise consider the dual function $\neg f(\bar{x}_1, \dots, \bar{x}_n)$ which is $n - k - 1$ -sensitive). Let $\mathbf{Z} = \{x_1, \dots, x_{k-1}\} \subset \mathbf{X}_n$ and consider the function, $g(\mathbf{X}_n - \mathbf{Z}) = f^{\mathbf{Z} := \mathbf{1}}(\mathbf{X}_n - \mathbf{Z})$ of $n - k + 1$ variables. Obviously $\mathbf{L}(f) \geq \mathbf{L}(g)$. Since f is k -sensitive and $k - 1$ variables have been fixed to 1 in order to obtain g , it follows that $g(\mathbf{X}_n - \mathbf{Z})$ is 1-sensitive. We claim that $g(\mathbf{X}_n - \mathbf{Z})$ does not have a 3-refinement. To see this

suppose the contrary and that $\mathbf{Y} = \{y_1, y_2, y_3\}$ is a subset of $\mathbf{X}_n - \mathbf{Z}$ for which,

$$g^{\mathbf{X}_n - \mathbf{Z} - \mathbf{Y} := 0}(\mathbf{Y}) \equiv (y_1 \oplus y_2 \oplus y_3) \theta (y_1 \vee y_2 \vee y_3) \quad (4.3)$$

for some $\theta \in B_2$.

But $g^{\mathbf{X}_n - \mathbf{Z} - \mathbf{Y} := 0}(\mathbf{Y})$ is still 1-sensitive and so its value when exactly one y_i is 1 should differ from its value when all three y_i 's are 1. Obviously the right-hand side of (4.3) does not satisfy this requirement; hence $g(\mathbf{X}_n - \mathbf{Z})$ does not have a 3-refinement. Applying Corollary(4.4) we have,

$$\mathbf{L}(g(\mathbf{X}_n - \mathbf{Z})) \geq \varepsilon (n - k + 1) \log \log(n - k + 1)$$

Since $k \leq n/2$ and $\mathbf{L}(f) \geq \mathbf{L}(g)$ this gives

$$\mathbf{L}(f(\mathbf{X}_n)) \geq \delta n \log \log n \quad \square$$

Corollary 4.5: For all but 16 symmetric functions $f \in S_n$, $\mathbf{L}(f) \geq \delta n \log \log n$.

Proof: From Thm(4.11) if $\mathbf{L}(f) < \delta n \log \log n$ and f is symmetric then it must be the case that f is not k -sensitive for any $1 \leq k \leq n - 3$. If $\underline{w}(f) = w_0 w_1 \cdots w_{n-3} w_{n-2} w_{n-1} w_n$ is the spectrum of f , this implies that $w_k = w_{k+2}$ for each $1 \leq k \leq n - 3$ thus there are exactly 4 possible settings of bits $w_1 \cdots w_{n-1}$ which define non k -sensitive symmetric functions. For each of these w_0 and w_n may be set arbitrarily. It follows that there are exactly $4 \cdot 2 \cdot 2 = 16$ symmetric functions which are not k -sensitive. \square

The 16 symmetric functions referred to arise by setting c_1, c_2, c_3 and c_4 in the function below:

$$c_1 \oplus (c_2 \wedge (\bigoplus_{i=1}^n x_i)) \oplus (c_3 \wedge \bigwedge_{i=1}^n x_i) \oplus (c_4 \wedge (\bigvee_{i=1}^n x_i))$$

all of which clearly have linear formula size.

4.2.3) The Fischer/Meyer/Paterson Lower Bound

Pudlak (1983) relates $L(f)$ to the cardinality of the maximal subset \mathbf{Y} of \mathbf{X}_n for which,

$$f^{\mathbf{X}_n - \mathbf{Y} := 0}(\mathbf{Y}) \equiv \left(\bigoplus_{y \in \mathbf{Y}} y \right) \theta \left(\bigvee_{y \in \mathbf{Y}} y \right)$$

(Fischer et al., 1982) consider assignments to $\mathbf{X}_n - \mathbf{Y}$ which fix nearly equal numbers of variables to 0 and 1 and relate $L(f)$ to the size of the largest subset, \mathbf{Y} of \mathbf{X}_n , for which:

\exists a partial assignment π to $\mathbf{X}_n - \mathbf{Y}$ such that,

FMP1)

$$0 \leq |\{i : x_i^\pi := 1\}| - |\{i : x_i^\pi := 0\}| \leq 1$$

FMP2)

$$f^{\mathbf{X}_n - \mathbf{Y} := \pi}(\mathbf{Y}) \text{ is affine.}$$

As with the lower bounds presented earlier, the results described in this section are for formulae over the basis B_2 . In fact it is convenient to make use of the following fact which simplifies a number of proofs involved.

Fact 4.4: Let $F(\mathbf{X}_n)$ be any formula over the basis B_2 . There exists a formula $G(\mathbf{X}_n)$ over the basis $\{\wedge, \oplus, 0, 1\}$ equivalent to F and such that: $\forall x \in \mathbf{X}_n, \text{occ}(x, G) \leq \text{occ}(x, F)$. Equality holding in the case of F containing only \wedge -type and \oplus -type gates.

Proof: Instances of projections or constant functions may be eliminated from F by applying the result of Lemma(1.3)(ii), so without loss of generality we may assume that F contains only \wedge -type and \oplus -type gates. It is now sufficient to construct an equivalent formula G

over $\{\wedge, \oplus, 0, 1\}$ satisfying $occ(x, G) = occ(x, F)$. It is easy to see that any subformula $F_1 \theta F_2$ of F , where $\theta \notin \{\wedge, \oplus\}$ may be replaced by a subformula,

$$\left\{ \begin{array}{l} ((F_1 \oplus \alpha) \wedge (F_2 \oplus \beta)) \oplus \gamma \text{ if } \theta \text{ is } \wedge\text{-type} \\ \alpha \oplus F_1 \oplus F_2 \text{ if } \theta \text{ is } \oplus\text{-type} \end{array} \right\}$$

where α, β, γ are constants. Clearly the number of occurrences of x is not increased by any such replacement and this proves the result. \square

From this fact it follows that we may concentrate on deriving lower bounds for $occ(\mathbf{X}_n, G)$, G as above, in producing lower bounds on \mathbf{L} , viz $\mathbf{L}(f) = occ(\mathbf{X}_n, F) - 1 = occ(\mathbf{X}_n, G) - 1$, F an optimal formula realising f and G its equivalent $\{\wedge, \oplus, 0, 1\}$ -representation.

Below we use π, σ, τ (possibly subscripted) to denote partial assignments to \mathbf{X}_n . $dom(\pi) \subseteq \mathbf{X}_n$ is the set of variables fixed by π , so that $|\pi| = |dom(\pi)|$. The *eccentricity* of π , $ecc(\pi)$ is the quantity

$$|\{x \in dom(\pi) : x^{|\pi} := 1\}| - |\{x \in dom(\pi) : x^{|\pi} := 0\}|$$

π is *central* if $ecc(\pi)$ is 0 or 1. σ is an *extension* of π if and only if $dom(\pi) \subseteq dom(\sigma)$ and for any $x \in dom(\pi)$ we have $x^{|\pi} = x^{|\sigma}$. $dom(\sigma, \pi) = dom(\sigma) - dom(\pi)$ are the additional variables fixed by σ in extending π .

F, G and H (again possibly subscripted) will denote formulae over the basis $\{\wedge, \oplus, 0, 1\}$. $var(F)$ is the set of variables which are used in F , note that this may be a superset of the variables upon which the function represented by F essentially depends. The *dimension of F* , $dim(F)$, is defined to be $|var(F)|$. F is said to be *affine* if and only if F realises an affine Boolean function. G is an *affine variant* of F if the formula $F \oplus G$ is affine. As previously, F is an *r-formula* if $occ(x, F) \leq r$ for all $x \in var(F)$ and is *r-minimal* with respect

to some property of formulae, if $occ(var(F), F)$ is minimal among r -formulae with the desired property.

Given F and π , the *restriction of F w.r.t π* , $F|_{\pi}$ is that formula obtained by replacing each instance of $x \in dom(\pi) \cap var(F)$ by the constant $x|_{\pi}$. If π is central and $dom(\pi) \subseteq var(F)$ then $F|_{\pi}$ is a *central restriction of F* . Finally the *affine diameter of F* , $diam(F)$, is the maximal dimension of any affine central restriction of F .

The lower bound theorem proved by (Fischer et al., 1982) asserts that there is some constant $\varepsilon > 0$ with which any formula F having $var(F) = \mathbf{X}_n$ satisfies

$$occ(\mathbf{X}_n, F) \geq \varepsilon n \log \left(\frac{n}{diam(F)} \right)$$

Important consequences of this result are lower bounds on specific symmetric functions, such as MAJ_n , of $\Omega(n \log n)$ and an alternative lower bound on the formula size of k -sensitive functions which improves Thm(4.11) for certain values of k .

Following the presentation of (Fischer et. al, 1982) we describe this result in three stages. The core of the argument is contained in the Main Lemma below which, with the aid of four preliminary results whose proof forms the initial stage, establishes that given any F and central π one can construct an central extension σ of π such that $F|_{\sigma}$ is affine and has dimension closely related to $dim(F|_{\pi})$. Given this, the lower bound theorem is relatively easily verified.

Lemma 4.5: (The Affine Variant Lemma) Let G be any affine variant of F . Then for all π

- i) $G|_{\pi}$ is affine $\Leftrightarrow F|_{\pi}$ is affine.

- ii) If G is an r -minimal affine variant of F , for some $r \geq 1$ and $\text{dom}(\pi) \subseteq \text{var}(G)$ then

$$\dim(F_{|\pi}) - \dim(G_{|\pi}) = \dim(F) - \dim(G)$$

Proof: Since $F \oplus G$ is affine and any restriction is also affine it follows that $(F \oplus G)_{|\pi} \equiv F_{|\pi} \oplus G_{|\pi}$ is affine. Thus if $G_{|\pi}$ is affine then so is the formula,

$$F_{|\pi} \oplus G_{|\pi} \oplus G_{|\pi} \equiv F_{|\pi} \oplus 0 \equiv F_{|\pi}$$

and this proves part(i). To prove part(ii) simply observe that $\text{var}(G) \subseteq \text{var}(F)$ for otherwise we can replace each variable in $\text{var}(G) - \text{var}(F)$ by a constant in G to yield a smaller affine variant r -formula of F . The relation now follows easily from the facts that

$$\dim(G_{|\pi}) = \dim(G) - |\pi| \quad ; \quad \dim(F_{|\pi}) = \dim(F) - |\pi| \quad \square$$

Lemma 4.6: (The Conjunction Lemma) Let π be central and F such that $F_{|\pi} = G \wedge H$ where G and H are affine. There is a central extension σ of π for which $\text{dom}(\sigma, \pi) \subseteq \text{var}(F_{|\pi})$, $F_{|\sigma}$ is affine, and $\dim(F_{|\sigma}) \geq \dim(F_{|\pi})/3$.

Proof: We may express G and H respectively as,

$$G \equiv \bigoplus_{u \in U} u \oplus \bigoplus_{w \in W} w \oplus c$$

$$H \equiv \bigoplus_{v \in V} v \oplus \bigoplus_{w \in W} w \oplus d$$

where $W = \text{var}(G) \cap \text{var}(H)$, $U = \text{var}(G) - W$, $V = \text{var}(H) - W$ and c, d are constants.

Now if $\sigma_1, \sigma_2, \sigma_3$ are central extensions of π which add $U \cup W, V \cup W$ and $U \cup V$ respectively to $\text{dom}(\pi)$ then each formula $F_{|\sigma_i}$ is affine. Since

$$\bigcup_{i=1}^3 \text{var}(F_{|\sigma_i}) = U \cup V \cup W = \text{var}(F_{|\pi})$$

it follows that one of these restrictions has the required dimension. \square

Lemma 4.7: (The Partition Lemma) Let S_1, S_2, \dots, S_t be a collection of sets and $T = \bigcup_{1 \leq i < j \leq t} S_i \cap S_j$. There exists a partition of $\{1, 2, \dots, t\}$ into two sets A and B such that

$$|P \cap Q| = |(\bigcup_{i \in A} S_i) \cap (\bigcup_{j \in B} S_j)| \geq |T|/2$$

Proof: Let $x \in T$ so that $x \in S_i \cap S_j$ for some $1 \leq i < j \leq t$. We say that a partition (A, B) splits $x \in S_i \cap S_j$ if and only if $i \in A$ and $j \in B$ (or vice-versa). Clearly at least 2^{t-1} partitions split x and so the average size of $P \cap Q$ is at least,

$$\frac{\sum_{x \in T} |\{(A, B) : (A, B) \text{ splits } x\}|}{2^t}$$

which is $\geq |T|/2$. It follows that at least one partition has the required property. \square

The final preliminary lemma introduces a function whose properties will be essential to the inductive proof of the main lemma.

Lemma 4.8: (The Beta Lemma) There are constants $\alpha > 0$, $a > 1$ such that the function $\beta: \mathbf{N} \rightarrow \mathbf{R}$ given by $\beta(r) = (\alpha a^r C_r)^{-1}$, C_r being the Catalan number $\binom{2r-2}{r-1}/r$, satisfies:

- i) $\beta(r) = \frac{\alpha}{\sum_{s=1}^{r-1} (\beta(s) \beta(r-s))^{-1}}$ for $r > 1$.
- ii) $\beta(r) \leq (1 - 15\alpha)/6 < 1$ for $r \geq 1$.

iii) $\beta(r) \leq (1 - 5\alpha)/(1 - 5\alpha + 4r)$.

Proof: (i) follows from the fact that $C_r = \sum_{s=1}^{r-1} C_s C_{r-s}$, (Knuth, 1973, pp. 388-9). (ii) and (iii) hold since $C_r \sim d r^{-3/2} 4^r$ for some constant $d > 0$ (using Stirling's approximation), thus for small enough α one can find a constant a to satisfy (ii) and (iii), e.g. $\alpha = 1/30$, $a = 360$. \square

The lower bound theorem will follow from the main lemma below.

Lemma 4.10: Let F be an r -formula ($r \geq 1$) and π_0 be central. There is a central extension, π of π_0 such that $F|_{\pi}$ is affine; $\text{dom}(\pi, \pi_0) \subseteq \text{var}(F)$ and

$$\dim(F|_{\pi}) \geq \beta(r) \dim(F|_{\pi_0})$$

Proof: By course-of-values on induction on r . Thus assume $r \geq 1$ and that the result holds for all r' -formulae with $r' < r$. To establish the lemma for all r -formulae we apply a course-of-values subinduction on $\text{occ}(\text{var}(F), F)$. The subinductive base being immediate, let F be any r -formula, π_0 any central assignment and assume that the lemma holds for all r -formulae G , having $\text{occ}(\text{var}(G), G) < \text{occ}(\text{var}(F), F)$.

First observe that we may assume F to be an r -minimal affine variant of $F|_{\pi_0}$ and hence $F = F_{\pi_0}$ and $\text{var}(F) \cap \text{dom}(\pi_0) = \emptyset$. To see this suppose that there is an r -minimal affine variant, G , of $F|_{\pi_0}$ having

$$\text{occ}(\text{var}(G), G) < \text{occ}(\text{var}(F), F)$$

Then applying the subinduction hypothesis we find a central extension π of π_0 satisfying the lemma for G . From the affine variant lemma (i) it follows that $F|_{\pi}$ is affine and in addition,

$$\dim(F|_{\pi}) = \dim(G|_{\pi}) + (\dim(F|_{\pi_0}) - \dim(G)) \quad \text{AVL (ii)}$$

$$\begin{aligned}
&\geq \beta(r) \dim(G) + (\dim(F|_{\pi_0}) - \dim(G)) && \text{Induction} \\
&\geq \beta(r) \dim(F|_{\pi_0})
\end{aligned}$$

This last from the Beta Lemma (ii), since $\beta(r) < 1$.

Thus if a smaller affine variant, G , exists then the central extension π that establishes the result for G also establishes the result for F . As $F|_{\pi_0}$ is obviously an affine variant of itself and an r -formula then such a G is guaranteed to exist *unless*

$$\text{occ}(\text{var}(F|_{\pi_0}), F|_{\pi_0}) = \text{occ}(\text{var}(F), F)$$

i.e. $\text{var}(F) \cap \text{dom}(\pi_0) = \emptyset$ so that $F = F|_{\pi_0}$.

The lemma holds for F if and only if it holds for $1 \oplus F$ so without loss of generality we may write F as,

$$F = \bigoplus_{i=1}^k F_i$$

No F_i can be affine, otherwise $\bigoplus_{j \neq i} F_j$ is a smaller affine variant of $F = F|_{\pi_0}$ and this contradicts the assumption that F is r -minimal amongst such affine variants. It follows that each F_i is of the form $G_i \wedge H_i$, where neither G_i nor H_i are equivalent to constant functions.

Now consider the following partition of $\text{var}(F_i)$ into 4 sets:

$$\begin{aligned}
\text{global}(F_i) &= \text{var}(F_i) \cap \left(\bigcup_{j \neq i} \text{var}(F_j) \right) \\
\text{joint}(G_i, H_i) &= (\text{var}(G_i) \cap \text{var}(H_i)) - \text{global}(F_i) \\
\text{own}(G_i) &= \text{var}(G_i) - (\text{joint}(G_i, H_i) \cup \text{global}(F_i)) \\
\text{own}(H_i) &= \text{var}(H_i) - (\text{joint}(G_i, H_i) \cup \text{global}(F_i))
\end{aligned}$$

Clearly $\text{var}(F) = \mathbf{global} \cup \mathbf{joint} \cup \mathbf{own}$ where

$$\begin{aligned}\mathbf{global} &= \bigcup_{i=1}^k \text{global}(F_i) \\ \mathbf{joint} &= \bigcup_{i=1}^k \text{joint}(G_i, H_i) \\ \mathbf{own} &= \bigcup_{i=1}^k (\text{own}(G_i) \cup \text{own}(H_i))\end{aligned}$$

and these three sets are pairwise disjoint.

The proof of the Main lemma is completed by a case analysis involving the size of these sets in relation to $n = \text{dim}(F) = \text{dim}(F|_{\pi_0})$.

Case 1: $n \leq 1/\beta(r)$. Choose any central extension, π of π_0 for which $\text{dom}(\pi, \pi_0) \subset \text{var}(F)$ and $\text{dim}(F_\pi) = 1$. Any formula in a single variable is affine and by the assumption that $n \leq 1/\beta(r)$ we have

$$\text{dim}(F_\pi) = 1 \geq \beta(r)n = \beta(r)\text{dim}(F|_{\pi_0})$$

as required.

Case 2: $|\mathbf{global}| \geq 2\alpha n$. \mathbf{global} is the set of variables of F which occur in at least two of the sets $\text{var}(F_i)$. Applying Lemma(4.7) to the collection $\text{var}(F_1), \dots, \text{var}(F_k)$ yields a partition (A, B) of $\{1, 2, \dots, k\}$ such that

$$\left| \bigcup_{i \in A} \text{var}(F_i) \cap \bigcup_{i \in B} \text{var}(F_i) \right| \geq |\mathbf{global}|/2 \geq \alpha n$$

F is equivalent to the formula $P \oplus Q$ where

$$P = \bigoplus_{i \in A} F_i ; Q = \bigoplus_{i \in B} F_i$$

Consider the set of variables in $\text{var}(P) \cap \text{var}(Q)$. Each of these occurs fewer than r times in both P and Q . For $1 \leq s < r$ let V_s denote the subset of those variables in $\text{var}(P) \cap \text{var}(Q)$ for which

$occ(V_s, P) = s$, hence $occ(V_s, Q) \leq r - s$. We have,

$$\sum_{s=1}^{r-1} |V_s| = |var(P) \cap var(Q)| \geq \alpha n$$

From the Beta Lemma (i),

$$\alpha n = \sum_{s=1}^{r-1} \frac{\beta(r) n}{\beta(s) \beta(r-s)}$$

So combining these it follows that for some t , $1 \leq t \leq r - 1$ it holds,

$$|V_t| \geq \frac{\beta(r) n}{\beta(t) \beta(r-t)}$$

We intend to apply the main lemma, inductively, to some restriction of P , Q to V_t . Let σ be any central extension of π_0 having $dom(\sigma, \pi_0) = var(F) - V_t$ and P' , Q' be the formulae $P|_\sigma$, $Q|_\sigma$ respectively. Thus $var(P') = var(Q') = V_t$, P' is a t -formula and Q' an $(r-t)$ -formula. From the inductive hypothesis, for r , there is a central extension τ of σ such that $P'|_\tau$ is affine and

$$\begin{aligned} dim(Q'|_\tau) &= dim(P'|_\tau) \geq \beta(t) \cdot dim(P) \\ &= \beta(t) \cdot |V_t| \geq \frac{\beta(r) n}{\beta(r-t)} \end{aligned}$$

$Q'|_\tau$ is an $(r-t)$ -formula so again applying the inductive hypothesis for r gives a central extension π of τ for which $(Q'|_\tau)|_\pi = Q'|_\pi$ is affine and

$$dim(Q'|_\pi) \geq \beta(r-t) dim(Q'|_\tau) \geq \beta(r) n$$

Observing that,

$$(P \oplus Q)|_\pi = (P'|_\tau)|_\pi \oplus Q'|_\pi$$

is affine from the construction of τ and π , and that $\dim((P \oplus Q)_{|\pi}) = \dim(Q'_{|\pi}) \geq \beta(r)n$ completes the proof of Case(2).

Case 3: $|\mathbf{joint}| \geq 3\alpha n$. Any variable in $x \in \mathbf{joint}$ satisfies for some i : x occurs in F_i only and

$$1 \leq \max\{occ(x, G_i), occ(x, H_i)\} \leq r-1$$

Let $u_i = |\mathbf{joint}(G_i, H_i)|$. Following an argument used in Case(2) we can, for each i , find a value t_i ($1 \leq t_i \leq r-1$) such that, V_i being the set of variables in $\mathbf{joint}(G_i, H_i)$ for which $occ(V_i, G_i) = t_i$, satisfies

$$|V_i| \geq \frac{u_i \beta(r)}{\alpha \beta(t_i) \beta(r-t_i)}$$

Since $occ(V_i, G_i) = t_i$, clearly $occ(V_i, H_i) \leq r-t_i$. We now apply the inductive hypothesis to identify a suitable extension of π_0 .

Let σ_0 be any central extension of π_0 having

$$\text{dom}(\sigma_0) = \text{var}(F) - \bigcup_{i=1}^k V_i$$

Furthermore let $F'_i = (F_i)_{|\sigma_0}$, $G'_i = (G_i)_{|\sigma_0}$ and $H'_i = (H_i)_{|\sigma_0}$. An extension satisfying the requirements of the lemma is constructed in k steps. At the i 'th step we find a central extension, σ_i of σ_{i-1} , such that $(F'_i)_{|\sigma_i}$ is affine and

$$\dim((F'_i)_{|\sigma_i}) \geq u_i \beta(r)/(3\alpha)$$

For step i consider G'_i . This is a t_i -formula and $\text{var}(G'_i) = V_i$. Applying the inductive hypothesis to G'_i and σ_{i-1} , yields a central extension π_i of σ_{i-1} such that $(G'_i)_{|\pi_i}$ is affine and

$$\dim((G'_i)_{|\pi_i}) \geq \beta(t_i) \dim(G'_i)$$

Applying the inductive hypothesis to H'_i , an $(r - t_i)$ -formula, and π_i , we find a central extension τ_i of π_i such that $(H'_i)_{|\tau_i}$ is affine and

$$\dim((H'_i)_{|\tau_i}) \geq \beta(r - t_i) \dim(H'_i)$$

Now $(G'_i)_{|\tau_i}$ is also affine, since the restriction of an affine function is still affine, so from Lemma(4.6), the Conjunction Lemma, there is a central extension σ_i of τ_i for which $(F'_i)_{|\sigma_i}$ is affine and

$$\dim((F'_i)_{|\sigma_i}) \geq \dim((F'_i)_{|\tau_i})/3$$

From the choice of σ_0 ,

$$\text{var}(F'_i) = \text{var}(G'_i) = \text{var}(H'_i) = V_i$$

Hence,

$$\begin{aligned} \dim((H'_i)_{|\tau_i}) &\geq \beta(r - t_i) \dim((H'_i)_{|\pi_i}) = \beta(r - t_i) \dim((G'_i)_{|\pi_i}) \\ &\geq \beta(r - t_i) \beta(t_i) |V_i| \\ &\geq u_i \beta(r)/\alpha \end{aligned}$$

Thus,

$$\dim((F'_i)_{|\sigma_i}) \geq \dim((H'_i)_{|\tau_i})/3 \geq u_i \beta(r)/(3\alpha)$$

If we set $\pi = \sigma_k$, the assignment at the end of the k 'th step, then $F_{|\pi} = \bigoplus_{i=1}^k (F'_i)_{|\sigma_i}$ and thus is affine. In addition

$$\begin{aligned} \dim(F_{|\pi}) &\geq \sum_{i=1}^k \dim((F'_i)_{|\sigma_i}) \geq \sum_{i=1}^k u_i \beta(r)/(3\alpha) \\ &= |\mathbf{joint}| \beta(r)/3\alpha \end{aligned}$$

which is at least $\beta(r)n$ by the condition on $|\mathbf{joint}|$. This establishes Case(3).

Case 4: $|\mathbf{own}| \geq (1 - 5\alpha)n$ and $n > 1/\beta(r)$. If none of Cases(1-3) hold then the condition of Case(4) must be satisfied. We will construct a central extension, σ of π_0 for which $\text{dom}(\sigma) \subseteq \text{var}(F)$ and such that the function realised by $F_{|\sigma}$ is independent of some (non-empty) subset, V , of $\text{var}(F_{|\sigma})$. For such an assignment let yield denote the size of the associated V and $\text{cost} = \text{yield} + \text{dom}(\sigma, \pi_0)$. It will be sufficient to construct σ so that $\text{yield} \geq \beta(r) \text{cost}$, for then we can find a central extension, π of σ , which satisfies the lemma for F .

To see that $\text{yield} \geq \beta(r) \text{cost}$ implies the existence of π consider the formula $G = (F_{|\sigma})_{|\tau}$ where τ is any assignment to V . Since $F_{|\sigma}$ is functionally independent of V we have $G \equiv F_{|\sigma}$. In addition

$$\text{var}(F) = \text{dom}(\sigma, \pi_0) \cup V \cup \text{var}(G)$$

and these three sets are disjoint. Thus $\text{dim}(F) = \text{cost} + \text{dim}(G)$. By the condition that $V \neq \emptyset$, τ must fix at least one variable of $F_{|\sigma}$ hence $\text{occ}(\text{var}(G), G) < \text{occ}(\text{var}(F), F)$. Applying the subinduction hypothesis for G and σ we find a central extension π such that $G_{|\pi}$ is affine, $\text{dom}(\pi, \sigma) \subseteq \text{var}(G)$ and $\text{dim}(G_{|\pi}) \geq \beta(r) \text{dim}(G)$. Now, $G \equiv F_{|\sigma}$ and since π extends σ it follows that $G_{|\pi} \equiv F_{|\pi}$ therefore $F_{|\pi}$ is affine. Furthermore $\text{dom}(\pi) \cap V = \emptyset$, by choice of G , so

$$\text{var}(F_{|\pi}) = V \cup \text{var}(G_{|\pi})$$

Thus,

$$\begin{aligned} \text{dim}(F_{|\pi}) &= \text{yield} + \text{dim}(G_{|\pi}) \\ &\geq \beta(r) \text{dim}(G) + \beta(r) \text{cost} = \beta(r) \text{dim}(F) \end{aligned}$$

It follows that it is only necessary to identify σ and V having $yield / cost \geq \beta(r)$ in order to complete the proof.

Let $g_i = |own(G_i)|$ and $h_i = |own(H_i)|$. Since \wedge is commutative, without loss of generality we may assume that $g_i \geq h_i$ for each $1 \leq i \leq k$ and hence $\sum_{i=1}^k g_i \geq |own|/2$.

For each i there are 2 possible ways in which we can prevent F 's dependence on a subset of $own(G_i)$ or $own(H_i)$. Either we can find a central extension, σ of π_0 , which renders H_i 0, and hence F becomes independent of $var(G_i)$; or we can find one which renders H_i 1 and hence F becomes independent of $var(H_i) - dom(\sigma, \pi_0)$. In the former case we wish to minimise the participation of $own(G_i)$ in constructing σ the better to maximise $yield$; in the latter we wish to minimise the contribution from $own(H_i)$ for the same reasons. These alternatives give rise to 2 possible strategies.

Strategy A: Applicable only if there is a central extension, σ of π_0 , with $dom(\sigma, \pi_0) \subseteq var(F)$ and such that $(H_i)_{|\sigma} \equiv 0$. Find a minimal (number of variables) central extension, σ of π_0 , such that

$$\begin{aligned} (H_i)_{|\sigma} &\equiv 0 \\ var(H_i) &\subseteq dom(\sigma) \subseteq var(F) \\ &\text{and for which} \\ |dom(\sigma) \cap own(G_i)| \end{aligned}$$

is as small as possible amongst all such extensions. With this approach $F_{|\sigma}$ is functionally independent of $V = own(G_i) - dom(\sigma)$

Strategy B: Applicable only if there is a central extension, σ of π_0 , with $dom(\sigma, \pi_0) \subseteq var(H_i)$ and for which $(H_i)_{|\sigma} \equiv 1$. Find a maximal (number of variables) subset V of $own(H_i)$ such that there is a central extension σ satisfying $(H_i)_{|\sigma} \equiv 1$ and

$dom(\sigma, \pi_0) = var(H_i) - V$. $F|_\sigma$ is functionally independent of V ; the cost is $|V|$; the yield $dim(H_i)$.

We will show that for some i at least one of these strategies delivers a *yield/cost* value of the required magnitude. Recall that no H_i is equivalent to a constant function and so there is certainly an extension τ of π_0 for which $(H_i)|_\tau \equiv 0$. Let $\delta(H_i)$ denote the smallest integer such that there is an extension τ of π_0 having $dom(\tau, \pi_0) = var(H_i)$, $(H_i)|_\tau \equiv 0$ and

$$-\delta(H_i) \leq ecc(\tau) \leq \delta(H_i) + 1$$

Note that $\delta(H_i) \leq dim(H_i)$; the "+ 1" term is necessary in the case where τ is already central and $dim(H_i)$ is odd.

Suppose Strategy(A) is applicable with σ being the relevant central extension. Let τ be such that

$$dom(\tau) = dom(\pi_0) \cup var(H_i)$$

and for any $x \in dom(\tau)$ it holds that $x|^\tau = x|^\sigma$. Now σ is a minimal central extension which nullifies H_i so from the definition of δ we must have $ecc(\tau) = -\delta(H_i)$ or $\delta(H_i) + 1$. Thus the set $dom(\sigma, \tau)$ contains the minimal number of variables needed to extend τ to a central assignment, i.e

$$|dom(\sigma, \pi_0)| = dim(H_i) + \delta(H_i)$$

Clearly $own(G_i) - dom(\sigma) \subseteq var(F) - dom(\sigma)$; if this inclusion were strict then there would be some variable in $var(F) - dom(\sigma)$ which could have been used in making τ central. σ is chosen to utilise as few variables as possible from $own(G_i)$ so it follows that either $dom(\sigma) \cap own(G_i) = \emptyset$ or $own(G_i) - dom(\sigma) = var(F) - dom(\sigma)$. Therefore,

$$yield_A = |own(G_i) - dom(\sigma)| = \min(g_i, n - dim(H_i) - \delta(H_i))$$

and

$$\begin{aligned} cost_A &= dim(H_i) + \delta(H_i) + yield_A \\ &= \min(g_i + dim(H_i) + \delta(H_i), n) \end{aligned}$$

In the event of Strategy(A) not being applicable we set $cost_A = n$ and $yield_A = 0$ so that these expressions continue to be valid.

If for some $1 \leq i \leq k$ we have $\frac{yield_A}{cost_A} \geq \beta(r)$ then Strategy(A) is successful.

Now consider any application of Strategy(B), where V is the associated redundant set of variables. We claim that $|V| \geq \min(h_i, \delta(H_i) - 1)$. For let $V' \subseteq own(H_i)$ with $|V'| = \min(h_i, \delta(H_i) - 1)$. Furthermore let σ be any central extension of π_0 having $dom(\sigma, \pi_0) = var(H_i) - V'$ and τ be any assignment to V' . We have,

$$\begin{aligned} -\delta(H_i) &< -\min(h_i, \delta(H_i) - 1) && \text{obviously} \\ &= -|dom(\tau)| \\ &\leq ecc(\sigma \cup \tau) && \text{By centrality of } \sigma \\ &\leq |dom(\tau)| + 1 && \text{By centrality of } \sigma \\ &= \min(h_i, \delta(H_i) + 1) < \delta(H_i) + 1 \end{aligned}$$

From the definition of δ , it must be the case that $(H_i)_{\sigma \cup \tau} \equiv 1$. This is so regardless of the choice of τ and therefore $(H_i)_{\sigma} \equiv 1$ and $(H_i)_{\sigma}$ is functionally independent of V' . Strategy(B) chooses V , with these same properties, as large as possible so certainly $|V| \geq |V'| = \min(h_i, \delta(H_i) - 1)$ as claimed.

So for Strategy(B) we have,

$$yield_B \geq \min(h_i, \delta(H_i) - 1) \quad ; \quad cost_B = dim(H_i)$$

If Strategy(B) does not apply set $yield_B = 0$ and $cost_B = dim(H_i)$ so that these continue to hold.

Again if $\frac{yield_B}{cost_B} \geq \beta(r)$, for some i , then Strategy(B) is successful.

It is now shown, by contradiction, that there is some i for which either Strategy(A) succeeds or Strategy(B) succeeds.

Suppose neither strategy is successful. Since (A) fails $\frac{yield_A}{cost_A} < \beta(r)$. So for each $1 \leq i \leq k$,

$$(1 - \beta(r)) \min(g_i + dim(H_i) + \delta(H_i), n) \leq dim(H_i) + \delta(H_i) \quad (4.4)$$

(B) also fails, so for each i , $\frac{yield_B}{cost_B} \leq \beta(r)$, hence

$$\min(h_i, \delta(H_i) - 1) \leq \beta(r) dim(H_i) \quad (4.5)$$

Let $\mu = |\mathbf{global} \cup \mathbf{joint}| \leq 5\alpha n$. Summing over i and applying the Case(4) premise gives,

$$\delta(H_i) \leq dim(H_i) \leq \mu + h_i \leq n - \sum_{j=1}^k g_j$$

$$\leq n - \frac{|\mathbf{own}|}{2} \leq (1 + 5\alpha)n/2 \quad (4.6)$$

From (4.5) and (4.6) we have,

$$\delta(H_i) - \mu - 1 \leq \min(h_i, \delta(H_i) - 1) < \beta(r) \dim(H_i) \leq \beta(r)n \quad (4.7)$$

From (4.6), (4.7), the fact that $\beta(r)n > 1$ and part (ii) of the beta lemma, we get,

$$\begin{aligned} \dim(H_i) + \delta(H_i) &\leq (1 + 5\alpha)n/2 + m + 1 + \beta(r)n \\ &< (1 + 15\alpha)n/2 + 2\beta(r)n \leq (1 - \beta(r))n \end{aligned} \quad (4.8)$$

If the "min" in (4.4) is n then (4.8) is contradicted. Hence the first argument is always less than n . Now (4.4) gives for each $1 \leq i \leq k$,

$$(1 - \beta(r))g_i < \beta(r)(\dim(H_i) + \delta(H_i)) \leq 2\beta(r)\dim(H_i) \quad (4.9)$$

and so,

$$\begin{aligned} (1 - \beta(r))(1 - 5\alpha)n/2 &\leq (1 - \beta(r))|\mathbf{own}|/2 \\ &\leq (1 - \beta(r)) \sum_{i=1}^k g_i \\ &< 2\beta(r) \sum_{i=1}^k \dim(H_i) < 2\beta(r)rn \end{aligned} \quad (4.10)$$

since F is an r -formula. (4.10) contradicts part (iii) of the Beta Lemma and so for some i Strategy(A) or Strategy(B) must succeed.

This establishes the last case and the lemma. \square

Theorem 4.12: (Fischer, Meyer, Paterson; 1982) There exists a constant $\varepsilon > 0$ with which any Boolean formula, F , having $\text{var}(F) = \mathbf{X}_n$ satisfies

$$\text{occ}(\mathbf{X}_n, F) \geq \varepsilon n \log \left(\frac{n}{\text{diam}(F)} \right)$$

Proof: Let F be any formula with $\text{var}(F) = \mathbf{X}_n$. Fix r equal to $\lfloor 2 \text{occ}(\mathbf{X}_n, F)/n \rfloor$ and let π_0 be a central assignment for which,

$$\text{dom}(\pi_0) = \{x : \text{occ}(x, F) > r\}$$

$F|_{\pi_0}$ is an r -formula so from Lemma(4.10) there is a central extension, π of π_0 for which $F|_{\pi}$ is affine; $\text{dom}(\pi, \pi_0) \subseteq \text{var}(F)$ and

$$\text{dim}(F|_{\pi}) \geq \beta(r) \text{dim}(F|_{\pi_0}) \quad (4.11)$$

With the same argument that commenced the proof of Theorem(4.10) we have,

$$\text{dim}(F|_{\pi_0}) \geq n/2 \quad (4.12)$$

Furthermore, from the asymptotic approximation to C_r , given in the proof of the Beta Lemma, we can find some (large) $K > 1$ with which,

$$\beta(r) \geq \frac{2}{K^r} \quad (4.13)$$

Combining (4.11), (4.12) and (4.13) gives

$$\text{dim}(F|_{\pi}) \geq \left(\frac{2}{K^r} \right) \cdot \left(\frac{n}{2} \right) = \frac{n}{K^r}$$

which, solved in terms of r , yields

$$r \geq \frac{\log(n/\dim(F|_{\pi}))}{\log K} \quad (4.14)$$

Therefore,

$$\begin{aligned} \text{occ}(\mathbf{X}_n, F) &\geq r n/2 && \text{by choice of } r \\ &\geq \varepsilon n \log(n/\dim(F|_{\pi})) && \text{by (4.14), with } \varepsilon = 1/(2 \log K) \\ &\geq \varepsilon n \log(n/\text{diam}(F)) && \text{by definition of } \text{diam}(F) \quad \square \end{aligned}$$

Corollary 4.6: For all $f(\mathbf{X}_n) \in B_n$,

$$\mathbf{L}(f) \geq \varepsilon n \log\left(\frac{n}{\text{diam}(f)}\right) - 1$$

$\text{diam}(f)$ is the natural extension of diam from formulae to functions. \square

Corollary(4.6) leads to an alternative version of Thm(4.11).

Lemma 4.11: Let $f \in B_n$ be $\lfloor n/2 \rfloor$ -sensitive. Then

$$\mathbf{L}(f) \geq \varepsilon n \log(n/2)$$

Proof: It is sufficient to show that $\text{diam}(f) \leq 2$ for any $\lfloor n/2 \rfloor$ -sensitive f . Suppose the contrary and that there is a central restriction of f , π say, such that $|\text{dom}(\pi)| = n - 3$ and for which $f|_{\pi}(x_i, x_j, x_k) \equiv c \oplus x_i \oplus x_j \oplus x_k$. Since π is central exactly $\lceil (n-3)/2 \rceil$ variables in $\text{dom}(\pi)$ are set to 1, thus $f|_{\pi}$ must be 1-sensitive. This contradicts the assumption that $f|_{\pi}$ is affine. The lower bound is now immediate from Corollary(4.6). \square

Theorem 4.13: Let $f \in B_n$ be k -sensitive where $0 \leq k \leq n - 2$. Then

$$\mathbf{L}(f) \geq \varepsilon n \log \min(k, n - k)$$

Proof: Let f be k -sensitive for some k satisfying the theorem conditions. As in Thm(4.11) it may be assumed that $k \leq \lfloor n/2 \rfloor$. Let π be any partial assignment for which $|\text{dom}(\pi) \cap \mathbf{X}_n| = n - 2k$ and such that $x^{|\pi} = 0$ for each $x \in \text{dom}(\pi)$. Then $f^{|\pi} \in B_{2k}$ and is k -sensitive. From Lemma(4.11) it follows that,

$$\mathbf{L}(f^{|\pi}) \geq \varepsilon 2k \log k$$

At least one of the variables in $\mathbf{X}_n - \text{dom}(\pi)$ must occur $\geq \varepsilon \log k$ times in a minimal formula, F realising f , since one must occur this often in the restricted formula $F_{|\pi}$. In such a formula F , choose $\text{dom}(\pi)$ to be the $n - 2k$ most frequently occurring variables in F . From the previous argument it follows that with this choice $\forall x \in \text{dom}(\pi) \text{ occ}(x, F) \geq \varepsilon \log k$. Thus,

$$\mathbf{L}(f) \geq (n - 2k)\varepsilon + \mathbf{L}(f^{|\pi}) \geq \varepsilon n \log k \quad \square$$

Assuming $k \leq \lfloor n/2 \rfloor$ and comparing this result with Thm(4.11) we see that Pudlak's methods give larger bounds for k -sensitive functions whenever $k = o(\log^r n)$ for all $r > 0$, whereas Fischer, Meyer and Paterson's approach is superior for $k = \omega(\log^r n)$ for all $r > 0$. In the case $k = \theta(\log^r n)$ for some $r > 0$ both techniques give asymptotically equal bounds.

Two particular classes of interest are the functions T_k^n and C_k^n . For these we have,

$$\mathbf{L}(T_k^n) = \max \{ \Omega(n \log k), \Omega(n \log \log n) \} \quad 2 \leq k \leq \lfloor n/2 \rfloor$$

$$\mathbf{L}(C_k^n) > \varepsilon n \log(n/k) \quad \text{For } k \text{ fixed}$$

4.3) Formula size and depth

At the start of Section(2.3) we outlined some results concerning the depth complexity of functions with respect to formula size over various bases. In particular Theorem(2.13) shows that for any formula of size \mathbf{L} it is possible to construct an equivalent formula of depth $O(\log \mathbf{L})$. One undesirable side-effect of the transformation from a formula F to a formula G of depth $O(\log \mathbf{L}(F))$ is that $\mathbf{L}(G)$ may be $\omega(\mathbf{L}(F)^2)$ using the existing constructions. Examples of such behaviour are given by Pratt (1975a) for the algorithm of Spira (1971a).

In this section we present a result from Commentz-Walter (1979) which proves that for a specific family of functions decreasing depth involves a compensating increase in formula size. Thus for these functions there do not exist formulae which have simultaneously minimal depth and optimal size. The results apply only to the monotone basis $\{\wedge, \vee\}$. For the unate basis $\{\wedge, \vee, \neg\}$ Commentz-Walter and Sattler (1980) have proved a similar result for the same family of functions. Given the additional technical complexity of the latter we will be content to present only the monotone trade-off result in full. Both are derived by obtaining a lower bound on the product $\mathbf{L}(f) \cdot \mathbf{D}(f)$.

Consider the following family of functions, f_n , defined over disjoint sets of Boolean variables $\mathbf{X}_n = \langle x_1, \dots, x_n \rangle$ and $\mathbf{Y}_n = \langle y_1, \dots, y_n \rangle$

$$f_1 = y_1 \wedge x_1 \quad ; \quad f_n = y_n \wedge (x_n \vee f_{n-1})$$

Equivalently,

$$f_n = \bigvee_{i=0}^{n-1} (x_{n-i} \wedge \bigwedge_{j=n-i}^n y_j)$$

It is not difficult to see that f_n may be realised by a formula, over the basis $\{\wedge, \vee\}$, having size $2n$ and depth $2n-1$. On the other hand Commentz-Walter (1979) has shown that f_n may also be realised by a monotone formula of depth $O(\log n)$ but with size $O(n \log n)$. In the following sections we examine the complexity measure, \mathbf{P}_Ω ; the minimal $Size \times Depth$ of a formula for $f \in B_n$, i.e

$$\mathbf{P}_\Omega(f) = \min \{ \mathbf{L}_\Omega(F) \cdot \mathbf{D}_\Omega(F) : F \text{ is an } \Omega\text{-formula for } f \}$$

Throughout this section we consider only formulae over the basis $\{\wedge, \vee\}$ and subsequently \mathbf{P} instead of $\mathbf{P}_{\{\wedge, \vee\}}$ is used.

The aim of this section is to prove,

$$\mathbf{P}(f_n) = \Omega(n \log^2 n)$$

Notation: It is assumed that formulae realising f_n are constructed from the basis of arbitrary fan-in \wedge and \vee gates. This is convenient for considering formulae of constant depth. With the preceding assumption our measure of formula size will be $occ(\mathbf{X}_n \cup \mathbf{Y}_n, F)$ for such a formula F . For a formula F , $Ind(F)$ is the set of indices of literals occurring in F , e.g. if $F = (x_1 \vee y_2) \wedge x_2$ then $Ind(F) = \{1, 2, \}$. For $M \subseteq Ind(F)$, \mathbf{Z}_M denotes the set of variables,

$$\bigcup_{i \in M} \{x_i, y_i\}$$

$occ(i, F)$ denotes $occ(x_i, F) + occ(y_i, F)$. It will also be convenient to consider the average number of occurrences of any variable in a formula F . This we denote by $rel(var(F), F)$ and is equal to $\frac{occ(var(F), F)}{|var(F)|}$. When $var(F) = \mathbf{X}_n \cup \mathbf{Y}_n$ we will use simply $occ(n, F)$ and $rel(n, F)$.

A partial assignment, π , is said to be *reducing* if for each f_n it holds: after renaming of variables $f_n^{\pi} = f_m$, for some $m \leq n$. Let σ_i , where $1 \leq i \leq n$, be such that $\text{dom}(\sigma_i) = \{x_i, y_i\}$ and,

$$\sigma_i(x_i) = 0 \ ; \ \sigma_i(y_i) = 1$$

It is easily verified that, for each $1 \leq i \leq n$,

$$f_n^{\sigma_i} = f_{n-1}(\mathbf{X}_n - \{x_i\}, \mathbf{Y}_n - \{y_i\}) \quad (4.15)$$

So σ_i is a reducing assignment. Additionally since $\text{dom}(\sigma_i) \cap \text{dom}(\sigma_j) = \emptyset$ whenever $i \neq j$ for any set, I , of indices one may define σ_I as $\bigcirc_{i \in I} \sigma_i$; the assignment constructed by composing each of the assignments σ_i , for $i \in I$. This is also a reducing assignment.

The key idea in proving the lower bound on $\mathbf{P}(f_n)$ is to consider upper bounds on the measure, t defined as

$$\max \{ n : \exists F \text{ realising } f_n \text{ s.t. } \mathbf{D}(F) \leq d, \text{rel}(F) \leq s/2 \}$$

Lemma 4.12: There is a constant c such that for all $d \geq 1, s \geq 1$:

$$\log t \leq \sqrt{c d s}$$

We defer the proof of this lemma, first showing how it is applied to yield the cited trade-off.

Theorem 4.14: For all $n \geq 1, \mathbf{P}(f_n) = \Omega(n \log^2 n)$

Proof: Consider any monotone formula, F , realising f_n which is optimal with respect to $\mathbf{P}(f_n)$. For some $d \geq 1$ and $s \geq 1$ it holds that $\mathbf{D}(F) = d$ and $(s-1)/2 \leq \text{rel}(F) \leq s/2$ hence $n \leq t$ by definition. From Lemma(4.12) we now have

$$\begin{aligned} \mathbf{P}(f_n) &= \text{occ}(F) \cdot \mathbf{D}(F) = 2n \text{rel}(F) \mathbf{D}(F) \\ &\geq n(s-1)d = \Omega(n \log^2 n) \quad \square \end{aligned}$$

It remains to prove Lemma(4.12). This is accomplished in two stages; first we consider a simpler measure, $t'(d, s)$, defined as

$$\max \left\{ n : \exists F \text{ realising } f_n \text{ s.t. } \mathbf{D}(F) \leq d \text{ and } \text{occ}(\{x_i, y_j\}, F) \leq s \right\}$$

It is shown that t and $t'(d, s)$ are closely related. Finally an upper bound on $t'(d, s)$ is proved which will be of sufficient magnitude to deduce Lemma(4.12). This last part is the most involved section of the proof.

Before embarking on the proof of Lemma(4.12) we require some preliminary results on the structure of monotone formulae realising f_n .

Fact 4.5: Let δ be the partial assignment which fixes x_1 and y_n to the constant 1. Then

$$f_n^{|\delta} \equiv \tilde{f}_{n-1}(y_1, \dots, y_{n-1}, x_2, \dots, x_n)$$

Proof: The result follows easily from the expanded definition of f_n using elementary Boolean manipulations. \square

Fact 4.6: Let $m : \mathbf{N} \rightarrow \mathbf{R}^+$ be given by,

$$m(n) = \max \left\{ \binom{d+s}{s} : ds \leq n, d, s \in \mathbf{N} \right\}$$

Then for all $n \in \mathbf{N}$, $\log m(n) \leq \sqrt{cn}$ for some $c > 0$.

Proof: Omitted. \square

Lemma 4.13: Suppose that $\{g_0, \dots, g_l\}$ is a set of monotone Boolean functions over $\mathbf{Z}_n = \mathbf{X}_n \cup \mathbf{Y}_n$ for which $f_n = \bigvee_{i=0}^l g_i$. There is a partition of the indices $[1, \dots, n]$ into $l+1$ sets, I_0, \dots, I_l and a set of $l+1$ partial assignments $\{\pi_0, \dots, \pi_l\}$ having $\text{dom}(\pi_h) = \{x_j, y_j : j \notin I_h\}$ such that

$$\text{i) } g_h^{\upharpoonright \pi_h} = f_{|I_h|}.$$

Furthermore, there is a permutation Π of $\{0, \dots, l\}$ with which

$$\text{ii) } \text{For each } h=0, \dots, l \text{ and } i=1, \dots, n \text{ if } i \in I_h \text{ then } y_i \in \text{var}(g_k^{\upharpoonright \pi_k}) \text{ for all } k \text{ with } \Pi(k) \leq \Pi(h).$$

Proof: Let $V = \{1, \dots, n\}$ denote the index set of f_n and p_i denote the prime implicant $x_i \wedge \bigwedge_{j=i}^n y_j$ of f_n . Since $f_n = \bigvee_{j=0}^l g_j$ we have that,

$$\{p_1, \dots, p_n\} \subseteq \bigcup_{j=0}^l \mathbf{PI}(g_j) \subseteq \mathbf{I}(f_n) \quad (4.16)$$

From this fact we can construct some mapping $\phi : \mathbf{PI}(f_n) \rightarrow [0 \dots l]$ which satisfies for each $1 \leq i \leq n$:

$$\phi(p_i) = j \iff p_i \in \mathbf{PI}(g_j).$$

If $\phi(p_i) = j$ we say that p_i is assigned to g_j . For each $0 \leq j \leq l$, ϕ affords a partition of $\mathbf{PI}(g_j)$ into 2 sets: those prime implicants of f_n which are assigned to g_j ; and the *additional* prime implicants. Note that from (4.16) every additional prime implicant of g_j is an implicant of f_n . From this it follows that for every additional prime implicant, q , there is some index i_q such that q is a lengthening of p_{i_q} . From the construction of ϕ and this last property we have: if q is an additional prime implicant of g_h and a lengthening of p_j then $\phi(p_j) \neq h$.

We can now construct the partition of the index set $[1 \cdots n]$ into $l + 1$ sets, I_h , as follows: for each $0 \leq h \leq l$, $I_h = \{i : \phi(p_i) = h\}$. Now let π_h be the partial assignment σ_{V-I_h} . With this assignment

$$\begin{aligned} f_{|I_h|} &= f_n^{|\pi_h|} \\ &= \bigvee_{j=0}^l g_j^{|\pi_h|} \equiv g_h^{|\pi_h|} \end{aligned}$$

for by our previous arguments, whenever $j \neq h$, $g_j^{|\pi_h|}$ has its assigned prime implicants rendered 0 under π_h , as also those additional prime implicants which are not lengthenings of p_i , for $\phi(p_i) = h$. In the same way all additional prime implicants of g_h also become 0 under π_h . We have thus proved part (i) of the lemma. For part(ii) it is sufficient to define Π as the permutation which sorts g_0, \dots, g_l in ascending order of their lowest index assigned prime implicant. \square

Lemma 4.14: For all $d, s \geq 1$: $t \leq 3t'(d, 6s)$.

Proof: By definition for each $d, s \geq 1$ we can construct a monotone formula, F , having depth at most d and $rel(F) \leq s/2$ and such that F realises f_n with $n = t$. Let F be such a formula over the variable set $V = \mathbf{X}_n \cup \mathbf{Y}_n$. Since $rel(F) \leq s/2$ we have $occ(V, F) \leq ns$. Consider the sets $X \subseteq \mathbf{X}_n$ and $Y \subseteq \mathbf{Y}_n$ of x_i (resp. y_i) variables which occur at most $3s$ times in F . Since $3s(n - |X|) \leq occ(V, F)$ we must have $|X| \geq 2n/3$. In the same way $|Y| \geq 2n/3$ also. Let I_X, I_Y be the sets of indices corresponding to the variables of X, Y and $I = I_X \cap I_Y$. Clearly, from the lower bound on the cardinalities of X and Y , $|I| \geq n/3$. Applying the reducing assignment, $\sigma_{\{1, \dots, n\} - I}$ to F yields a formula G , of depth at most d , which realises $f_{|I|}$. Moreover by the construction of I , each variable $z \in var(G)$ occurs no more than $3s$ times, hence for each pair of distinct indices $\{j, k\} \in I$ we have $occ(\{x_j, y_k\}, G) \leq 6s$. It follows that $t'(d, 6s) \geq |I| \geq n/3$ and this

proves Lemma(4.14). \square

Proof of Lemma(4.12): To prove that $\log t \leq \sqrt{c d s}$ it is sufficient, from Lemma(4.14) and Fact(4.6), to establish

$$t'(d, s) \leq \binom{d+s}{s} - 1 \quad (4.17)$$

For this induction on $d \geq 0$ is used. The inductive base $d = 0$ and $s \leq 1$ is obvious. So assume that (4.17) is valid for all values $\leq d - 1$ and all $s' \leq s - 1$. It will be shown that (4.17) is also valid for d and s .

Let $n = t'(d, s)$. By definition there is some monotone formula, F , realising f_n in depth at most d and having $\text{occ}(\{x_i, y_j\}, F) \leq s$ for all pairs of indices i, j . We consider two cases.

Case 1 $F = G_0 \vee G_1 \vee \dots \vee G_l$: Let g_h be the (monotone) function realised by the sub-formula G_h of F . Then

$$f_n = g_0 \vee g_1 \vee \dots \vee g_l$$

Applying the result of Lemma(4.13) we find $l+1$ reducing assignments, π_h ($0 \leq h \leq l$) and a partition of the $\text{Ind}(F)$ into $l+1$ sets I_h for which

$$f_n^{|\pi_h} = f_{n_h} = g_h^{|\pi_h}$$

n_h denoting $|I_h|$. We may assume, since \vee is commutative, that the permutation, Π , of Lemma(4.13)(ii) is the identity, i.e $\Pi(h) = h$.

Now since $\{I_0, \dots, I_l\}$ defines a partition of $\text{Ind}(F)$ clearly, $\sum_{h=0}^l n_h = n = t'(d, s)$. We further have, from the properties of Π , that $\text{occ}(\{x_i, y_j\}, G_h^{|\pi_h}) \leq s - h$. So since $G_h^{|\pi_h}$ realises f_{n_h} and has depth at most $d - 1$ it follows that,

$$n_h \leq t'(d-1, s-h) \quad \forall h \quad 0 \leq h \leq l$$

Case 2 $F = G_0 \wedge G_1 \wedge \dots \wedge G_l$: Apply the partial assignment δ of Fact(4.5) to F . After relabelling of variables the resulting formula realises \tilde{t}_{n-1} . Hence the dual formula (in which the final gate will be an $l+1$ input \vee) realises f_{n-1} . Using the same argument as Case(1) we find a partition as before satisfying $\sum_{h=0}^l n_h = n-1$ and $n_h \leq t'(d-1, s-h)$ for each $0 \leq h \leq l$.

Thus with both cases,

$$\begin{aligned} t'(d, s) = n &\leq \sum_{h=0}^l n_h + 1 \\ &\leq \sum_{h=0}^l t'(d-1, s-h) + 1 \\ &\leq \sum_{h=0}^l \left(\binom{d-1+s-h}{s-1} - 1 \right) + 1 \\ &\leq \sum_{h=0}^s \left(\binom{d-1+h}{d-1} - 1 \right) \quad s > 1 \\ &= \binom{d+s}{s} - 1 \end{aligned}$$

The last line can be readily proved by induction on s . This completes the proof of Lemma(4.12). \square

For the unate basis $\{\wedge, \vee, \neg\}$ Commentz-Walter and Sattler (1980) have also proved,

Theorem 4.15:

$$\mathbf{P}_{\{\wedge, \vee, \neg\}}(f_n) = \Omega\left(\frac{n \log n \log \log n}{\log \log \log \log n}\right) \quad \square$$

4.4) Upper Bounds on Formula size for symmetric functions

We have seen in Chapter(2) that combinational networks can be constructed to realise any symmetric Boolean function of n arguments using $O(n)$ gates and $O(\log n)$ depth. In the case of monotone networks, the monotone symmetric functions, (i.e threshold functions), T_k^n may also be realised in kn monotone gates for fixed k , and $O(n \log n)$ gates for non-constant k . For formulae, the methods of (Fischer et al. 1982) yield $\Omega(n \log n)$ lower bounds for specific symmetric functions, such as MAJ_n , whereas Pudlak (1983) obtains $\Omega(n \log \log n)$ bounds for T_k^n , for constant $k \geq 2$. For the basis $\{\wedge, \vee, \neg\}$, Krichevskii (1964) had earlier proved $\Omega(n \log n)$ lower bounds for such T_k^n .

In this section a number of upper bounds on formula size for various symmetric functions are presented. The specific functions considered are: C_k^n in the cases where $k = 2^p$ and $k = 3$; and finally monotone formulae for threshold functions.

Peterson (1978) exhibits an upper bound of $O((\log n)n^{3.33635\dots})$ for the formula size of any symmetric function using the basis B_2 . This is based on ideas similar to the upper bound derived with respect to combinational networks: efficient formulae to compute the binary representation of the number of 1s among the n -inputs are constructed. Copies of the resulting $\lceil \log n \rceil$ formulae, S_0, S_1, \dots, S_p are then used in conjunction with an appropriate "universal" formula to compute the required symmetric function. In this S_i is the formula for the i 'th bit of the binary representation of $\sum_{j=1}^n x_j$.

Lemma 4.15:

$$\mathbf{L}(C_k^n) = \begin{cases} O(n(\log n)^{p-1}) & k = 2^p \quad (a) \\ O(n^2) & k = 3 \quad (b) \\ O(n^{2.58}) & k = 7 \quad (c) \\ O(n^3) & k = 5, 15 \quad (d) \end{cases}$$

Proof: (a) is from (Fischer et al., 1982). We describe the construction for $k=4$ only, leaving the generalisation to arbitrary constant powers of two as an exercise. We further assume that $n = 2^r$ for some $r \geq 1$. For any assignment α to \mathbf{X}_n let

$$\underline{\sigma}(\alpha) = \sigma_r \sigma_{r-1} \cdots \sigma_i \sigma_{i-1} \cdots \sigma_1 \sigma_0(\alpha)$$

denote the binary expansion of the number of 1s in α . In this σ_0 is the least significant bit. Clearly $C_4^n(\alpha) = 1$ if and only if $\sigma_1(\alpha)$ and $\sigma_0(\alpha)$ both equal 0. So it will suffice to construct formula for these two Boolean functions. Partition \mathbf{X}_n into 2 disjoint sets of variables \mathbf{Y} and \mathbf{Z} each of size $n/2$. Then,

$$\sigma_0(\mathbf{X}_n) = \bigoplus_{i=1}^n x_i \quad ; \quad \sigma_1(x) = 0$$

$$\sigma_1(\mathbf{Y}, \mathbf{Z}) = \sigma_1(\mathbf{Y}) \oplus \sigma_1(\mathbf{Z}) \oplus (\sigma_0(\mathbf{Y}) \wedge \sigma_0(\mathbf{Z}))$$

Let $S_0(n)$, $S_1(n)$ denote the size of the resulting formulae for σ_0 , σ_1 . Then with these expressions,

$$S_0(n) = n - 1 \quad ; \quad S_1(n) = 2S_1(n/2) + n - 1$$

Hence,

$$S_0(n) = n - 1 \quad ; \quad S_1(n) = O(n \log n).$$

This now establishes (a) since $C_4^n \equiv \text{NOR}(\sigma_0, \sigma_1)$.

For (b) let $C_{3,l}^n \in B_n$ be defined by,

$$C_{3,l}^n(\mathbf{X}_n) \iff \sum_{i=1}^n x_i \equiv l \pmod{3}$$

As in (a) let \mathbf{X}_n be partitioned into disjoint, equal sized sets \mathbf{Y} and \mathbf{Z} with n again a power of 2. Also let $q = (2l) \pmod{3}$; $r = (2l+1) \pmod{3}$ and $s = (2l+2) \pmod{3}$. It may be confirmed that,

$$C_{3,l}^n(\mathbf{X}_n) = [C_{3,q}^{n/2}(\mathbf{Y}) \iff C_{3,q}^{n/2}(\mathbf{Z})] \wedge [C_{3,r}^{n/2}(\mathbf{Y}) \iff C_{3,s}^{n/2}(\mathbf{Z})]$$

Solving the recurrence relation for $\mathbf{L}(C_3^n)$ given by these expressions yields the result claimed.

The upper bounds (c) and (d) are proved in Van Leijenhorst (1987). \square

In the remainder of this section we are concerned solely with monotone formulae.

One consequence of the $O(n \log n)$ -size, $O(\log n)$ -depth monotone sorting network of (Ajtai et al., 1983) is the existence of polynomial size monotone formulae for all threshold functions, cf. Thm(2.4). In practice there are two drawbacks to this result: the proof is non-constructive; the degree of the bounding polynomial, for functions such as MAJ_n , is extremely large. The next result described establishes the existence of monotone formulae for MAJ_n of "small" polynomial size.

Theorem 4.16: (Valiant, 1984) For each $n = 2m$ there is a monotone formula realising MAJ_n in depth $O(\log n)$ and with size $O(n^{5.3})$.

Proof: The proof is non-constructive. The existence of a suitable formula is shown by considering a sequence of probability distributions

for monotone formulae with variable set \mathbf{X}_n .

$$A_0, A_1, A_2, \dots, A_{i-1}, A_i, \dots,$$

This sequence is defined in such a way that for t large enough a formula selected at random according to A_t computes MAJ_n with probability at least $1/2$. Since each formula having non-zero probability in A_t will have depth at most $2t$ and size at most 2^{2t} the theorem will follow if it can be shown that $t = c \log n$, for some suitable constant $c > 0$, is an appropriate choice. The distribution A_i is given by considering the following random generation of a formula $F \in A_i$. Below $\alpha = (3 - \sqrt{5})/2$.

- i) If $i=0$ then F is either a literal x_j or the constant 0. The probability of F being the former, for any x_j is $2\alpha/(2m-1)$. The probability of the later choice is $1 - 2\alpha/(2m-1)$.
- ii) If $i > 0$ then F is formed by selecting formulae G_1, G_2, G_3 and G_4 independently according to the distribution A_{i-1} . F is then defined as the formula $(G_1 \vee G_2) \wedge (G_3 \vee G_4)$.

Now let π_0, π_1 be assignments to \mathbf{X}_n in which at most $m-1$, resp. at least m , variables are assigned the value 1. For a formula F chosen according to A_i let,

$$\begin{aligned} f_i &= \text{Prob} [F_{|\pi_0} \equiv 1] \\ h_i &= \text{Prob} [F_{|\pi_1} \equiv 0] \end{aligned}$$

To prove the theorem it suffices to establish that for some constant $c > 0$ and $t = c \log n$ we have

$$f_t < \frac{1}{2^{n+1}} \quad ; \quad h_t < \frac{1}{2^{n+1}} \quad (4.18)$$

For then, choosing a formula, F according to A_t , we have

$$Prob [F \neq MAJ_n] = \sum_{\pi \in \{0,1\}^n} Prob [F_{|\pi} \neq MAJ_n(\pi)]$$

and from (4.18) this summation is certainly less than $1/2$.

From the definition of the distributions A_i it is immediate that:

$$f_0 \leq \frac{2\alpha(m-1)}{2m-1} = \alpha - \frac{\alpha}{n-1} \tag{4.19}$$

$$h_0 \leq 1 - \frac{2\alpha m}{2m-1} = 1 - \alpha - \frac{\alpha}{n-1} \tag{4.20}$$

$$f_i = f_{i-1}^4 - 4 f_{i-1}^3 + 4 f_{i-1}^2 \tag{4.21}$$

$$h_i = -h_{i-1}^4 + 2 h_{i-1}^2 \tag{4.22}$$

We wish to find t for which $f_t < \frac{1}{2^{n+1}}$ and $h_t < \frac{1}{2^{n+1}}$. Suppose we know that for some $j \geq 0$, and some $\varepsilon > 0$, e.g $\varepsilon = 2^{-4}$, it holds that

$$f_j < \varepsilon \quad ; \quad h_j < \varepsilon$$

Let $t > j$ and $k = 2^{t-j}$. From (4.21) and (4.22) and the fact that $0 \leq f_{i-1}, h_{i-1} < 1$, it is clear that

$$f_i < 4 f_{i-1}^2 \quad ; \quad h_i < 4 h_{i-1}^2$$

Hence,

$$f_t < 4^{k-1} (f_j)^{2^k} < 2^{-2k} \quad ; \quad h_t < 4^{k-1} (h_j)^{2^k} < 2^{-2k}$$

Hence if $f_j < 2^{-4}$, then f_t for $t = \lceil \log n \rceil + j$, is less than $\frac{1}{2^{n+1}}$ and similarly for h_t .

So from the previous argument the theorem follows if we can show that $f_j < 2^{-4}$ for some $j = c \log n$. (An identical argument will hold for h_j so only one analysis is given in detail).

Consider the behaviour of f_i as a function of f_{i-1} and that of h_i as a function of h_{i-1} . It is easy to show that,

$$f_i(\alpha) = \alpha ; h_i(1-\alpha) = 1-\alpha$$

Additionally, for any $0 \leq \varepsilon \leq \alpha$ it holds,

$$\begin{aligned} f_i(\alpha - \varepsilon) &= \sum_{r=0}^4 \left(\frac{\varepsilon^r}{r!} \right) \frac{\delta^r f_i}{\delta f_{i-1}} (\alpha) \\ &= \alpha - 4\alpha\varepsilon + \varepsilon^2(6\alpha+2) - \varepsilon^3(4\alpha-4) + \varepsilon^4 \end{aligned}$$

Here $\frac{\delta^r y}{\delta x}$ denotes the r 'th derivative of $y(x)$ with respect to x and $\frac{\delta^0 y}{\delta x}$ is taken to be $y(x)$.

From the above expansion we have that for $0 < \varepsilon < \alpha$, $f_i(\alpha - \varepsilon) < \alpha - \varepsilon$. In particular, since $f_0 \leq \alpha - \frac{\alpha}{n-1}$, it holds that

$$\forall \gamma < 4\alpha - \left(\frac{\alpha}{n-1} \right) \left[6\alpha - 2 + \left(\frac{\alpha}{n-1} \right) \left[4\alpha - 4 + \frac{\alpha}{n-1} \right] \right],$$

$$\forall i > 0, \forall 0 < \varepsilon < \frac{\alpha}{n-1}$$

$$f_{i-1} = \alpha - \varepsilon \quad \Rightarrow \quad f_i < \alpha - \gamma \varepsilon$$

Similarly

$$h_{i-1} = \alpha - \varepsilon \quad \Rightarrow \quad h_i < 1 - \alpha - \gamma \varepsilon$$

Combining this with (4.19) and (4.20) we conclude that,

$$f_i < \alpha - (\gamma)^i \left(\frac{\alpha}{n-1} \right)$$

$$h_i < 1 - \alpha - (\gamma)^i \left(\frac{\alpha}{n-1} \right)$$

It follows that for some $j = \frac{\log n}{\log \gamma} + O(1)$,

$$f_j < 2^{-4} \quad ; \quad h_j < 2^{-4}$$

So for some constant $\beta \geq 0$ and any $t \geq \frac{\log n}{\log \gamma} + \log n + \beta$ it holds,

$$f_t \leq \frac{1}{2^{n+1}} \quad ; \quad h_t \leq \frac{1}{2^{n+1}}$$

Recalling that any formula with non-zero probability in A_t has size at most 2^{2^t} , we conclude that there exists a monotone formula realising MAJ_n of size $O(n^{2(1+\log_\gamma 2)}) = O(n^{5.3})$, proving the theorem. \square

Obviously Thm(4.16) implies the existence of $O(n^{5.3})$ size monotone formulae for all threshold functions, T_k^n . Results of Khrapchenko (1971a, 1971b) yield $\Omega(n^2)$ lower bounds on the monotone complexity of MAJ_n , see the next section. For k fixed substantially improved upper bounds may be proved, these matching the lower bound of Krichevskii (1964) cited earlier.

The *existence* of $O(n \log n)$ monotone formulae realising T_k^n was first established by Khasin (1969a). Khasin considered monotone formulae of the following form:

Let $n = pk$ and $\Delta = \langle \Pi_1, \Pi_1, \dots, \Pi_k \rangle$ be a partition of \mathbf{X}_n into k sets, each containing exactly p elements. The function

$$f_\Delta = \bigwedge_{j=1}^k \bigvee_{x \in \Pi_j} x$$

contains p^k prime implicants each with exactly k variables. Thus $\mathbf{PI}(f_\Delta) \subset \mathbf{PI}(T_k^n)$. Let us say that a prime implicant, m of T_k^n is *covered* by a partition Δ if $m \leq f_\Delta$. If $\Delta_1, \dots, \Delta_r$ is a set of r partitions of \mathbf{X}_n as above, then for some large enough r we have that $T_k^n \equiv \bigvee_{i=1}^r f_{\Delta_i}$. Khasin was unable to explicitly demonstrate that a specific set of $O(\log n)$ partitions would be suitable. However the following probabilistic argument was used to prove that such a set did exist.

Let $n = pk$ and $\langle f_1, f_2, \dots, f_i, \dots, \rangle$ be a sequence of monotone functions over \mathbf{X}_n defined as follows:

$f_1 = f_{\Delta_1}$, where Δ_1 is a partition of \mathbf{X}_n into k sets of size p . Δ_1 is selected at random from the set of all such partitions with probability $(p!)^k/n!$, i.e. each partition is equally likely. For $i > 1$, $f_i = f_{i-1} \vee f_{\Delta_i}$. Δ_i is chosen with probability $(p!)^k/n!$, independently of $\Delta_1, \dots, \Delta_{i-1}$.

Using $\mathbf{L}^m(f)$ to denote the size of the smallest monotone formula realising f , it is clear from the definition of f_Δ that, $\mathbf{L}(f_\Delta) = n - 1$. Given this we wish to show that for some $r = O(\log n)$, the probability that a function f_r , chosen as above, is equivalent to T_k^n is strictly greater than 0. If this holds then it is proved that,

$$\mathbf{L}^m(T_k^n) = O(n \log n) \quad \text{for } k \text{ fixed}$$

So consider any function f_r as defined. Clearly, since Δ_1, Δ_2 etc are selected independently,

$$\begin{aligned} \text{Prob} [f_r \neq T_k^n] &\leq \sum_{m \in \text{PI}(T_k^n)} \prod_{i=1}^r \text{Prob} [m \text{ not covered by } \Delta_i] \\ &\leq \binom{n}{k} (1 - \text{Prob} [x_1 \cdots x_k \text{ covered by } \Delta])^r \end{aligned}$$

where Δ is any partition chosen with probability $(p!)^k/n!$.

The last expression holds since for any pair of distinct prime implicants of T_k^n , s and t say, the number of partitions which cover s is equal to the number of partitions which cover t , so in the summation we may without loss of generality consider the prime implicant $x_1 \cdots x_k$.

How many partitions Δ cover $m = x_1 \cdots x_k$? Δ covers m if and only if each class Π_j of Δ contains exactly one x_i . If $x_{\sigma(j)}$ is the variable of m contained in Π_j , then σ is thereby a permutation of $[1 \cdots k]$. Also the sets $\Pi_j - \{x_{\sigma(j)}\}$ afford a partition of $n - k = (p - 1)k$ elements into k equal sized sets. It follows that the number of partitions which cover m is exactly $\frac{(n - k)! k!}{[(p - 1)!]^k}$ and thus,

$$\text{Prob} [\Delta \text{ covers } m] = \frac{(n - k)! k!}{[(p - 1)!]^k} \cdot \frac{(p!)^k}{n!} = p^k \cdot \binom{n}{k}^{-1}$$

In summary,

$$\text{Prob} [f_r \neq T_k^n] = \binom{n}{k} \left(1 - p^k \binom{n}{k}^{-1} \right)^r \leq \binom{n}{k} \exp \left(-r p^k \binom{n}{k}^{-1} \right)$$

r needs to be chosen large enough so that, $Prob[f_r \neq T_k^n] < 1$. So, we require

$$\exp\left(r p^k \binom{n}{k}^{-1}\right) > \binom{n}{k}$$

$$r > \binom{n}{k} p^{-k} \log_e \binom{n}{k}$$

Recalling that $n = pk$ and $\binom{n}{k} < n^k$ yields the condition,

$$r > k^{k+1} \log_e n = O(\log n) \quad \text{for } k \text{ fixed}$$

In total we have just proved,

Theorem 4.17: (Khasin, 1969a) For all fixed k , and all n there exists a monotone formula realising T_k^n of size $O(n \log n)$. \square

Following Khasin's results there were a number of attempts to find efficient constructive solutions. McColl (1977) derives formulae of size $O(n \log n (\log \log n)^{k-2})$ for $2 \leq k \leq 5$. Kleiman and Pippenger (1978) use an intricate technique to obtain formulae of size $O(n \log n \binom{k}{2}^{\log^* n})$. The problem of explicitly constructing monotone formulae of a size matching Khasin's existential bound was finally solved by Friedman (1986).

Friedman examined monotone formulae defined by constructing a sequence $\Delta_1, \dots, \Delta_r$ of sets of disjoint subsets of \mathbf{X}_n . Thus $\Delta_i = \langle \Pi_1^i, \Pi_2^i, \dots, \Pi_k^i \rangle$ the sets Π_s^i and Π_t^i being disjoint for $s \neq t$. Note that $\bigcup_{j=1}^k \Pi_j^i$ may be a strict subset of \mathbf{X}_n . We call any such Δ a

division of \mathbf{X}_n , to distinguish the possibility of Δ not being a strict partition. As in Khasin's example, monotone formulae, F , of size rn are used being defined as:

$$F = \bigvee_{i=1}^r \left(\bigwedge_{j=1}^k \bigvee_{x \in \Pi_j^i} x \right)$$

A concept of a division covering a prime implicant of T_k^n is defined as before and recalling the argument used earlier in deriving Khasin's bound we know that a division Δ covers $x_{i_1} \cdots x_{i_k}$ if and only if each x_{i_j} occurs in exactly one class Π of Δ . A set of divisions $\Delta_1, \dots, \Delta_r$ which collectively cover all prime implicants of T_k^n is called an (n, k) -scheme of size r .

As a preliminary stage consider the problem of constructing $(n, 2)$ -schemes of size $\lceil \log n \rceil$. It will be convenient to view the variables, \mathbf{X}_n , as $\{x_0, x_1, \dots, x_{n-1}\}$. Let $n = 2^r$ and for $0 \leq i \leq n-1$ let $Bin_j(i)$ denote the j 'th (most significant) bit of the r -bit binary expansion of i . Define the division Δ_j by,

$$\begin{aligned} \Pi_0^j &= \{x_i : Bin_j(i) = 0\} \\ \Pi_1^j &= \{x_i : Bin_j(i) = 1\} \end{aligned}$$

Now since for any two distinct x_i and x_j the binary expansions of i and j must differ in at least one digit it follows that $x_i x_j$ is covered by at least one division. So the set of $r = \lceil \log n \rceil$ divisions defined above gives rise to an $(n, 2)$ -scheme of size $\lceil \log n \rceil$ for any n .

Friedman notes that attempting to generalise this idea by representing indices in base k and defining divisions from commonality of the j 'th digit in the expansion breaks down since, with $k = 3$ for example, triples such as $\langle 222, 122, 221 \rangle = \{x_{26}, x_{17}, x_{25}\}$ have no

single position differing in all 3 expansions.

The key observation made in Friedman (1986) is that an (n, k) -scheme of size $O(\log n)$ can be constructed by building divisions based on variable indices represented in some base b , depending on k . Thus for each n and k there is a constant b_k with which: \exists a subset S of $\{1, 2, 3, \dots, b_k\}^m$ having size n and with the property that any k distinct elements of S differ in at least one position.

More formally we proceed as follows.

Let $\{1, 2, \dots, b\}^m$ denote the set of all m -tuples of the integers $\{1, 2, \dots, b\}$. For any pair of m -tuples $\alpha = \langle a_1, \dots, a_m \rangle$ and $\beta = \langle b_1, \dots, b_m \rangle$, the *Hamming distance*, $H(\alpha, \beta)$ is given as, $|\{i : a_i \neq b_i\}|$. The *ball of radius r about α* is the set of m tuples,

$$B_r(\alpha) = \{ \beta : H(\alpha, \beta) \leq r \}$$

Finally for any set, S of m -tuples, the *separation* of S is defined to be $\min \{ H(\alpha, \beta) : \alpha, \beta \in S, \alpha \neq \beta \}$.

Lemma 4.16: Let l be any integer greater than 1, $b = 2^{2l}$ and $c = 2l$. For all $m \in \mathbf{N}$ there exists $S \subset \{1, \dots, b\}^{mc}$ such that $|S| = b^m$ and having separation $> (1 - 1/l) mc$.

Proof: For $r \in \mathbf{N}$ and any mc -tuple α it is clear that

$$|B_r(\alpha)| \leq \binom{mc}{r} b^r$$

Hence for $\varepsilon = 1 - 1/l$ and any such α we have,

$$\begin{aligned} |B_{\varepsilon mc}(\alpha)| &\leq \binom{mc}{\varepsilon mc} b^{\varepsilon mc} \\ &< 2^{mc} b^{\varepsilon mc} = b^{(1-1/2l)mc} \end{aligned}$$

$$= \frac{|\{1, \dots, b\}^{mc}|}{b^m}$$

It follows that we can construct a suitable S as follows: $S = \{\alpha_1, \dots, \alpha_i, \dots, \alpha_{b^m}\}$. If $i = 1$ choose α_1 to be any mc -tuple in $\{1, \dots, b\}^{mc}$. If $i > 1$ α_i can be chosen as any tuple in the set,

$$\{1, 2, \dots, m\}^{mc} - \bigcup_{j=1}^{i-1} B_{\epsilon mc}(\alpha_j)$$

From the upper bound just proved, this set is non-empty while $i \leq b^m$. \square .

It should be noted that the set S in the preceding lemma is constructible in time polynomial in b^{mc} .

Theorem 4.18: (Friedman, 1986) For fixed k and any $n > k$ it is possible to construct (n, k) -schemes of size $O(\log n)$ in time polynomial in n .

Proof: Let $l = \binom{k}{2}$, $m = \lceil \log_b n \rceil$ and apply the preceding lemma to yield a set $S = \{x_1, \dots, x_{b^m}\}$. Consider the indices of k distinct points y_1, \dots, y_k in S . For any two different indices α, β in this set we know that these indices are $m2l$ -tuples and that $H(\alpha, \beta) > (1 - 1/l)2lm$. It follows that there is some i , $1 \leq i \leq 2lm$ such that the i 'th component of the k $2ml$ -tuples differs. We can thus construct a (b^m, k) -scheme of size $O(m)$ as follows:

For each j , $1 \leq j \leq mc$ and each

$$1 \leq t_1 < t_2 < \dots < t_k \leq b$$

the division $\Delta_{j, t_1, \dots, t_k}$ has its i 'th subset

$$\Pi_i^{j, t_1, \dots, t_k} =$$

given by

$$\{ p : j' \text{th component of } (2ml) \text{-tuple } x_p \text{ equals } t_i \}$$

The correctness of this construction is immediate from the preceding arguments and its size is,

$$mc \binom{b}{k} = O(m) \quad \square$$

4.5) Bounds for bases other than B_2

The previous section included some upper bounds on symmetric functions for the bases $\{\wedge, \vee, \neg\}$ and $\{\wedge, \vee\}$. In this final section we review some general lower bound techniques for these cases. The two methods examined are those of Khrapchenko (1971a, b), which improve some earlier work of Subbotovskaya (1961) concerning the power of the basis $\{\wedge, \vee, \neg\}$ and allow a lower bound on MAJ_n to be determined. The second method given is that of Andreev (1985) which is notable for being the largest bound on formula size over a complete basis attained to date, albeit for a somewhat artificial function. We note here that both techniques pertaining to the basis $\{\wedge, \vee, \neg\}$ exploit the absence of the operations \oplus and \iff and cannot be generalised to arbitrary bases.

4.5.1) The Khrapchenko Bound

In considering formulae over the basis $\{\wedge, \vee, \neg\}$ we may without loss of generality assume that negation is applied solely to the input nodes of a formula. This follows easily from De Morgan's Laws using the transformation of Lemma(3.32) and noting that there is no increase in the size of the formula because all gates have fanout equal

to 1.

Khrapchenko's lower bound is based on a measure defined for any non-constant Boolean function. Given any $f \in B_n$ we may define a partition of $\{0, 1\}^n$ into two sets of assignments:

$$\begin{aligned} f^{-1}(0) &= \{ \alpha \in \{0, 1\}^n : f(\alpha) = 0 \} \\ f^{-1}(1) &= \{ \alpha \in \{0, 1\}^n : f(\alpha) = 1 \} \end{aligned}$$

Let $NEXT_n$ denote the set of pairs of assignments $\langle \alpha, \beta \rangle$ for which the Hamming distance between α and β (i.e. $H(\alpha, \beta)$ using the notation of the previous section) is 1.

Theorem 4.19: (Khrapchenko, 1971a, b) Let $f \in B_n$ be a non-constant Boolean function, A be any non-empty subset of $f^{-1}(0)$, B any non-empty subset of $f^{-1}(1)$ and C be the set of pairs $A \times B \cap NEXT_n$. Then for all formulae F over the basis $\{\wedge, \vee, \neg\}$ realising f it holds,

$$occ(\mathbf{X}_n, F) \geq \frac{|C|^2}{|A| \cdot |B|}$$

Proof: The proof below is due to Paterson (pers. comm). Let F be any minimal (number of occurrences of literals) formula realising f over the basis $\{\wedge, \vee, \neg\}$. We proceed by induction on $occ(\mathbf{X}_n, F) \geq 1$ to prove the theorem.

The inductive base, $occ(\mathbf{X}_n, F) = 1$ is trivial. F is a formula of a single literal thus $f = x$ or $f = \neg x$. Therefore $|f^{-1}(0)| = |f^{-1}(1)| = 1$ and $|C| = 1$. It follows that $occ(x, F) = 1 = \frac{|C|^2}{|A| \cdot |B|}$ as required.

For the inductive step assume that $occ(\mathbf{X}_n, F) > 1$ and that the theorem holds for all smaller formulae. Since F contains at least two literals it follows that $F = G \theta H$ where $\theta \in \{\wedge, \vee\}$ and G and H are

smaller formulae. The case $\theta = \vee$ only is proved; the \wedge -gate case follows from a similar argument.

Let g, h be the functions realised by G and H respectively, so that $f = g \vee h$. Choose A and B so that $\frac{|C|^2}{|A| \cdot |B|}$ is maximised. Using A and B we wish to construct subsets, A_g, A_h, B_g and B_h of $g^{-1}(0), h^{-1}(0), g^{-1}(1)$ and $h^{-1}(1)$ in such a way that the inductive argument will succeed. Fix $A_g = A_h = A$; since $f = g \vee h$, f is 0 only for those assignments which render both g and h 0. Thus these choices are valid subsets of $g^{-1}(0)$ and $h^{-1}(0)$. Finally choose $B_g \subseteq B \cap g^{-1}(1)$ and $B_h \subseteq B \cap h^{-1}(1)$ in such a way that $B_g \cap B_h = \emptyset$ and $B_g \cup B_h = B$. Note that neither set is empty since F is chosen as a minimal formula. With these choices we have,

$$\begin{aligned} |A_g| &= |A_h| = |A| \\ |B| &= |B_g| + |B_h| \\ |C| &= |C_g| + |C_h| \end{aligned}$$

The last equality holds since B_g and B_h define a partition of B hence $C_g = (A \times B_g) \cap NEXT_n$ and $C_h = (A \times B_h) \cap NEXT_n$ define a partition of C . Now since $F = G \vee H$ we have that,

$$\begin{aligned} occ(\mathbf{X}_n, F) &= occ(\mathbf{X}_n, G) + occ(\mathbf{X}_n, H) \\ &\geq \frac{|C_g|^2}{|A| \cdot |B_g|} + \frac{|C_h|^2}{|A| \cdot |B_h|} \quad \text{By induction} \\ &\geq \frac{|C|^2}{|A| \cdot |B|} \end{aligned}$$

To see that the last line follows from its predecessor let $c_g = |C_g|$, $c_h = |C_h|$ etc and observe that the inequality asserted is

equivalent to

$$(c_g + c_h)^2 b_g b_h \leq (c_g^2 b_h + c_h^2 b_g)(b_g + b_h)$$

and this holds if and only if

$$(c_g b_h - c_h b_g)^2 \geq 0$$

Since this last condition is always satisfied the theorem follows. \square

Corollary 4.7: Let $PAR_n(\mathbf{X}_n) = \bigoplus_{i=1}^n x_i$. $\mathbf{L}_{\{\wedge, \vee, \neg\}}(PAR_n) \geq n^2$.

Proof: In Theorem(4.19) let $A = PAR_n^{-1}(0)$ and $B = PAR_n^{-1}(1)$. From the properties of \oplus , any two assignments α and β whose Hamming distance is 1 satisfy $PAR_n(\alpha) \neq PAR_n(\beta)$. It follows that $|A \times B \cap NEXT_n| = n|A| = n|B|$. The lower bound is now immediate from Thm(4.19). \square

Earlier Subbotovskaya (1961) had obtained a lower bound of $\Omega(n^{3/2})$ for the same function over this basis. Her methods are developed further in Andreev's lower bound below. It is obvious that PAR_n has formula size $n - 1$ over the basis B_2 and so these results show that no exact analogue of Lemma(1.4) can be proved for formula size over complete bases. It is known that the size can increase only polynomially in changing from one complete base to another (cf. Thm(2.13)). For the basis $\{\wedge, \vee, \neg\}$ Pratt (1975a) has established that for any $f \in B_n$ $\mathbf{L}_{\{\wedge, \vee, \neg\}}(f) \leq \mathbf{L}(f)^{\log_3 10}$. With the lower bound implied by Corollary(4.7) this exponent is close to optimal.

A further example of Khrapchenko's method is its application to threshold functions.

Corollary 4.8: $\mathbf{L}_{\{\wedge, \vee, \neg\}}(T_k^n) \geq k(n - k + 1)$.

Proof: Let

$$A = \{ \alpha : \alpha \text{ has exactly } k-1 \text{ 1's} \}$$

$$B = \{ \beta : \beta \text{ has exactly } k \text{ 1's} \}$$

Then every element of A is at Hamming distance 1 from exactly $n-k+1$ elements of B . Similarly every element of B is at Hamming distance 1 from exactly k elements of A . It follows that $|C| = (n-k+1)|A| = k|B|$ and this establishes the lower bound from Thm(4.19). \square

The bound of Corollary(4.8) is maximised for the majority function, $k = \lceil n/2 \rceil$ which has formula size $\geq n^2/4$. The best upper bound obtained to date, over this basis, is $O(n^5)$ using a universal symmetric function construction from Pippenger (1974).

4.5.2) The Andreev Bound

Andreev (1986) develops techniques of Subbotovskaya (1961) and Neciporuk (1966) to prove a lower bound of

$$\Omega\left(\frac{n^{5/2}}{(\log n)^{3/2} \log \log n}\right)$$

on the complexity of a specific n -input Boolean function when realised by the class of formulae over the basis $\{\wedge, \vee, \neg\}$.

Let

$$\left\{ \begin{array}{l} \tilde{x}_1 = (x_1^1, \dots, x_l^1) \\ \tilde{x}_2 = (x_1^2, \dots, x_l^2) \\ \dots \\ \dots \\ \tilde{x}_k = (x_1^k, \dots, x_l^k) \end{array} \right\}$$

be a set of disjoint tuples of Boolean variables. Let $\mathbf{X}_s^{k,l}$ denote the set of tuples, α of the form,

$$(j_{1,1}, \dots, j_{1,s}, \dots, j_{k,1}, \dots, j_{k,s}, \sigma_{1,1}, \dots, \sigma_{k,s})$$

such that

$$1 \leq j_{i,1} < j_{i,2} < \dots < j_{i,s} \leq l \quad \text{for } 1 \leq i \leq k$$

$$\sigma_{i,t} \in \{0, 1\} \quad \text{for } 1 \leq i \leq k; 1 \leq t \leq s$$

If $f(\tilde{x}_1, \dots, \tilde{x}_k)$ is a Boolean function then for $\alpha \in \mathbf{X}_s^{k,l}$, $f^{|\alpha}$ is the subfunction of f obtained by fixing $x_{j_{i,t}}^i = \sigma_{i,t}$ for each $1 \leq i \leq k$, $1 \leq t \leq s$.

Lemma 4.17: If $k \geq 1$, $l \geq 5$ then for any Boolean function $f(\tilde{x}_1, \dots, \tilde{x}_k)$ having $\mathbf{L}_{f \wedge, \vee, \neg}(f) \geq 2$, there exists $\alpha \in \mathbf{X}_1^{k,l}$ such that

$$\mathbf{L}_{f \wedge, \vee, \neg}(f^{|\alpha}) \leq \phi\left(\frac{1}{l}\right) \mathbf{L}_{f \wedge, \vee, \neg}(f)$$

where $\phi(x) = 1 - \frac{3x}{2} + \frac{x^2}{2}$

Proof: The proof is a probabilistic counting argument. Let ξ_f be a random variable $0 \leq \xi_f \leq \mathbf{L}_{f \wedge, \vee, \neg}(f)$ defined as follows: Randomly choose $\alpha \in \mathbf{X}_1^{k,l}$ with probability $(2l)^{-k}$, $\xi_f(\alpha)$ is the value $\mathbf{L}_{f \wedge, \vee, \neg}(f^{|\alpha})$.

We claim that the expected value of ξ_f , $E(\xi_f)$ is at most $\phi(1/l) \mathbf{L}_{f \wedge, \vee, \neg}(f)$. Clearly this is sufficient to prove the lemma. We use induction on $\mathbf{L}_{f \wedge, \vee, \neg}(f) \geq 2$ to establish this claim. If $\mathbf{L}_{f \wedge, \vee, \neg}(f) = 2$ the claim may be verified directly. Otherwise if $\mathbf{L}_{f \wedge, \vee, \neg}(f) > 2$ then $f \equiv f_1 \vee f_2$ or $f \equiv f_1 \wedge f_2$ for some Boolean functions $f_1 \neq f$, $f_2 \neq f$. Thus

$$\mathbf{L}_{l\wedge, \vee, \neg}(f) = \mathbf{L}_{l\wedge, \vee, \neg}(f_1) + \mathbf{L}_{l\wedge, \vee, \neg}(f_2)$$

If $\mathbf{L}_{l\wedge, \vee, \neg}(f_i) \geq 2$ for both $i = 1$ and $i = 2$ then by induction we have;

$$\begin{aligned} E(\xi_f) &\leq E(\xi_{f_1}) + E(\xi_{f_2}) \\ &\leq \phi\left(\frac{1}{l}\right) (\mathbf{L}_{l\wedge, \vee, \neg}(f_1) + \mathbf{L}_{l\wedge, \vee, \neg}(f_2)) \\ &\leq \phi\left(\frac{1}{l}\right) \mathbf{L}_{l\wedge, \vee, \neg}(f) \end{aligned}$$

Otherwise suppose that $\mathbf{L}_{l\wedge, \vee, \neg}(f_2) = 1$ so that $f_2 = (x_j^i)^\sigma \equiv x_j^i \oplus \sigma \oplus 1$. The function f_1 cannot essentially depend on x_j^i and this fact and the inductive hypothesis establish the claim in this case. \square

Lemma 4.18: There exists a positive constant c_0 such that if $k \geq 1$, $l > r \geq 4$ then for all $f(\tilde{x}_1, \dots, \tilde{x}_k)$ there exists $\alpha \in \mathbf{X}_{l-r}^{k,l}$ for which

$$\mathbf{L}_{l\wedge, \vee, \neg}(f^\alpha) \leq c_0 \left(\frac{l}{r}\right)^{-3/2} \mathbf{L}_{l\wedge, \vee, \neg}(f)$$

Proof: For $\mathbf{L}_{l\wedge, \vee, \neg}(f) \leq 1$ the result is immediate. Otherwise repeatedly applying the preceding lemma establishes the existence of $\alpha \in \mathbf{X}_{l-r}^{k,l}$ for which,

$$\mathbf{L}_{l\wedge, \vee, \neg}(f^\alpha) \leq \left(\prod_{m=r+1}^l \phi\left(\frac{1}{m}\right) \right) \mathbf{L}_{l\wedge, \vee, \neg}(f)$$

The value of the product in parenthesis is bounded above by $(l/r)^{-3/2}$ proving the result. \square

The following fact is immediate from this proof.

Lemma 4.19: If $k \geq 1$, $l > r \geq 4$, $g_1(\tilde{x}_1), \dots, g_k(\tilde{x}_k)$ do not become constant Boolean functions then for any integral r and

$$f = g(g_1(\tilde{x}_1), \dots, g_k(\tilde{x}_k)) \text{ holds,}$$

$$\mathbf{L}_{\{\wedge, \vee, \neg\}}(f) \geq \frac{1}{c_0} \left(\frac{l}{r}\right)^{3/2} \mathbf{L}_{\{\wedge, \vee, \neg\}}(g) \quad \square$$

Let k be any natural number $k \geq 3$ and $l = \lfloor \frac{2^k}{k} \rfloor$, $n = 2^k + kl$. The tuple \tilde{y} contains 2^k distinct variables $y_{\sigma_1 \dots \sigma_k}$, where $\sigma_i \in \{0, 1\}$ for each $1 \leq i \leq k$. These variables are distinct from $\tilde{x}_1, \dots, \tilde{x}_k$. Define $F_n(\tilde{y}, \tilde{x}_1, \dots, \tilde{x}_k)$ as

$$\bigvee_{\sigma_1, \dots, \sigma_k \in \{0,1\}} y_{\sigma_1 \dots \sigma_k} \wedge \left(\bigwedge_{i=1}^k \left(\bigoplus_{j=1}^l x_j^i \right)^{\sigma_i} \right)$$

and $\Phi_k(\tilde{y}, z_1, \dots, z_k)$ as

$$\bigvee_{\sigma_1, \dots, \sigma_k \in \{0,1\}} y_{\sigma_1 \dots \sigma_k} \wedge \left(\bigwedge_{i=1}^k z_i^{\sigma_i} \right)$$

Theorem 4.20:

$$\mathbf{L}_{\{\wedge, \vee, \neg\}}(F_n) = \Omega\left(\frac{n^{5/2}}{(\log n)^{3/2} \log \log n}\right)$$

Proof: Let E_m be the set of all binary tuples of length m and \hat{B}_k denote the set of all Boolean functions depending on k variables. Define

$$\mathbf{L}(k) = \max_{f \in \hat{B}_k} \mathbf{L}_{\{\wedge, \vee, \neg\}}(f)$$

For $\tilde{\lambda} \in E_{2^k}$ let,

$$\Phi_k^{\tilde{\lambda}} = \Phi_k(\tilde{\lambda}, z_1, \dots, z_k)$$

Clearly for any $\tilde{\lambda} \in E_{2^k}$

$$\mathbf{L}_{\{\wedge, \vee, \neg\}}(F_n) \geq \mathbf{L}_{\{\wedge, \vee, \neg\}}(\Phi_k^{\tilde{\lambda}}(\bigoplus_{j=1}^l x_j^1, \dots, \bigoplus_{j=1}^l x_j^k))$$

It follows from Lemma(4.19) that with $r = 4$ we have,

$$\mathbf{L}_{\{\wedge, \vee, \neg\}}(F_n) \geq \max_{\tilde{\lambda} \in E_{2^k}} \frac{1}{c_0} \left(\frac{l}{4}\right)^{3/2} \mathbf{L}_{\{\wedge, \vee, \neg\}}(\Phi_k^{\tilde{\lambda}})$$

Now since for any function $g(z_1, \dots, z_k) \in \hat{B}_k$ there exists $\tilde{\lambda} \in E_{2^k}$ such that $g \equiv \Phi_k^{\tilde{\lambda}}$, so we have

$$\begin{aligned} \mathbf{L}_{\{\wedge, \vee, \neg\}}(F_n) &\geq \frac{1}{c_0} \left(\frac{l}{4}\right)^{3/2} \mathbf{L}(k) \\ &\geq \left(\frac{2^k}{k}\right)^{3/2} \frac{2^k}{\log k} = \Omega\left(\frac{n^{5/2}}{(\log n)^{3/2} \log \log n}\right) \end{aligned}$$

□

Bibliographic Notes

Other lower bounds on formula size are given in Hodes (1970), Mehlhorn (1976) and Wechsung (1977). Bublitz (1986) considers monotone formulae for k -homogeneous functions. (Babai et al., 1987) prove $\Omega(n \log n)$ lower bounds on the size of (c, d) -formulae, i.e those permitting d -ary logic gates of fan-in c ; these results are obtained for certain symmetric functions.

A Boolean function $f \in B_n$ is said to essentially depend on m variables if for all subsets \mathbf{Y} of \mathbf{X}_n of size m and all assignments $\sigma \in \{0, 1\}^m$, $f^{\mathbf{Y} := \sigma}$ depends on all the variables $\mathbf{X}_n - \mathbf{Y}$. Malyshev (1967) proved that almost all Boolean functions essentially depend on m variables for $m \leq n - (1 + \varepsilon) \log n$, for any $\varepsilon > 0$ and n large enough. Using the approach of Subbotovskaya (1961) it is proved that for any f essentially dependent on m variables,

$$\mathbf{L}(f) \geq \max \left\{ \frac{n^{3/2}}{\sqrt{n-m}}, \frac{m \log m}{2 \log \log m} \right\}$$

Theorem(4.1) applied to monotone functions yields a lower bound of $\frac{2^n}{n^{1/2} \log n}$ on both monotone formula size and the size of formulae over B_2 . Red'kin (1979) gives constructions matching both lower bounds to within a constant factor. These employ ideas similar to those used in the proof of Theorem(3.4).

Bloniarz (1979) considers the formula size of $f(\mathbf{X}_n) \theta g(\mathbf{Y})$ for disjoint sets of variables \mathbf{X}_n and \mathbf{Y} and various $\theta \in B_2$. It is shown that formula size is additive, i.e $\mathbf{L}(f \theta g) = \mathbf{L}(f) + \mathbf{L}(g) + 1$.

Paul (1977) uses Neciporuk's lower bound method to construct a function with linear combinational complexity but formula size $\Omega(n^2/\log n)$.

Non-trivial lower bounds on depth have been obtained by McColl (1978c) for symmetric functions computed by the bases $\{\neg, \wedge\}$, $\{\neg, \wedge, \oplus\}$. These are bounds of $2 \lceil \log n \rceil$. McColl (1978a) presents a simple upper bound on the depth of monotone formulae. Recently, Karchmer and Wigderson (1987) have proved a $\Omega(\log^2 n / \log \log n)$ bound on the depth of monotone formulae computing transitive closure. Their argument relates monotone depth to a

measure of communication complexity and then uses information-theoretic arguments to produce the lower bound. Razborov (1988b) proves larger bounds of order $\Omega(\log^2 n)$ on the depth of monotone formulae for a set covering problem. Unfortunately a detailed presentation of these techniques would be too long to include here.

Chapter 5

Bounded-Depth Networks

... the order of an accidental series of accidents accidentally conceived.

Henry Miller

Tropic of Capricorn

5.1) Introduction

Bounded-depth networks allow arbitrary fan-in gates over the basis $\{\wedge, \vee\}$ but restrict depth to being constant. Formally

Definition 5.1: A depth- k network ($k \geq 0$) is a network which is a member of the class Σ_k or Π_k , these being defined inductively as follows.

i) If $k = 0$ then

$$\Sigma_0 = \Pi_0 = \{x_1, \dots, x_n, \bar{x}_1, \dots, \bar{x}_n, 0, 1\}$$

ii) If $k > 0$ then $S \in \Sigma_k$ if $S \in \Pi_{k-1}$ or S is formed by \vee -ing the outputs of a finite number of Π_{k-1} networks.

iii) If $k > 0$ then $S \in \Pi_k$ if $S \in \Sigma_{k-1}$ or S is formed by \wedge -ing the outputs of a finite number of Σ_{k-1} networks.

$BD_k^\Sigma(f)$ will denote the minimal size (number of wires) of any Σ_k network realising f . $BD_k^\Pi(f)$ is defined similarly. The depth- k complexity of f , denoted $BD^k(f)$ is given by

$$BD^k(f) = \min \{ BD_k^\Sigma(f), BD_k^\Pi(f) \} \quad \bullet$$

Note that for k constant we may interpret this definition as restricting gate fanout to be at most one, i.e dealing with *bounded-depth formulae*. This is because such a restriction only increases size polynomially and we will be interested only in bounds which are superpolynomial^{a)}.

One class of bounded-depth networks has already been encountered in Chapter(1). The representation of a Boolean function in *DNF* naturally defines a Σ_2 network (i.e a disjunction of conjunctions); the representation in *CNF* naturally gives rise to a Π_2 network (i.e a conjunction of disjunctions). Σ_k and Π_k may be seen as generalisations of these normal forms.

At first there seems to be little motivation for considering this class of network which was first introduced in Lupanov (1961a) and in fact lower bounds in this model give little insight into proof techniques relevant to combinational complexity. However its importance was demonstrated by (Furst et al., 1984) who established a connection between lower bounds on depth- k complexity and the *relativised polynomial-time hierarchy*.

Meyer and Stockmeyer (1973) introduced a hierarchy of complexity classes, lying between P and $PSPACE$, Σ_k^P and Π_k^P , consisting of languages over $\{0, 1\}^*$ characterised as follows.

$L \subseteq \{0, 1\}^*$ is in Σ_k^P if and only if all words $y \in L$ can be described as those satisfying an expression of the form,

$$\exists^P x_1 \forall^P x_2 \dots Q_k^P x_k R$$

a) Some authors define size as the total number of *gates* present. This again is polynomially equivalent to total number of wires.

where the quantifiers range over words (x_i) of length polynomial in $|y|$ and R is some decision problem in P . Π_k^P is defined as $co - \Sigma_k^P = \{0, 1\}^* - \Sigma_k^P$.

With this $P = \Sigma_0^P = \Pi_0^P$, $NP = \Sigma_1^P$, e.g the Directed Hamiltonian cycle problem of Chapter(3) may be expressed as: "Does there exist an ordering of the vertices (x_1) of a given directed graph (y) such that the ordering corresponds to a cycle of edges in y , (R) ?".

Since $PSPACE = \bigcup_{n=0}^{\infty} \Sigma_k^P \cup \Pi_k^P$, it is known that if a decision problem were found to be in Σ_{k+1}^P but not in Σ_k^P then this would enable a separation of P , NP and $PSPACE$ to be proved.

The problem of proving that the class of languages introduced above does indeed define a *proper* hierarchy is thus at least as difficult as proving $P \neq NP$ and $NP \neq PSPACE$.

The difficulty of resolving these issues led (Baker et al., 1975) to consider the apparently simpler question of whether separation could be achieved relative to an oracle.

An *oracle*, A , is just a subset of $\{0, 1\}^*$, i.e a language. A Turing machine, M , with oracle A has an additional "query" tape on which, at any stage during computation, M may consult the oracle A by writing a string, x and entering a "query" state. The answer to the question $x \in ? A$ determines the next move of M . The consultation of A is counted as a *single step* in the M 's computation.

Any complexity class, C , is extended in a natural way by the provision of an oracle A to a new complexity class C^A . C^A is said to be the class C *relativised* with respect to oracle A . In this way we can consider the question of whether there is an oracle A for which a separation of the classes of the polynomial-time hierarchy relativised with respect to A can be proven.

(Baker et al., 1975) successfully constructed an oracle A for which $P^A \subset NP^A$ could be demonstrated^{b)}. However, until recently, the best result obtained for a relativised hierarchy was an oracle A such that $\Sigma_2^{P,A} \neq \Pi_2^{P,A}$ and hence $\Sigma_2^{P,A} \subset \Sigma_3^{P,A}$, proved in (Baker and Selman, 1975). The techniques used therein did not seem powerful enough to separate other layers.

The significant breakthrough made by (Furst et al., 1984) was the discovery that an oracle, C , with which $\Sigma_k^{P,C} \subset \Sigma_{k+1}^{P,C}$, for all fixed $k \geq 2$, could be constructed if parity functions required exponential size depth- k networks, for all constant $k \geq 2$.

Lupanov (1961a) had earlier established that parity functions had exponential depth-2 complexity. (Furst et al., 1984) could achieve only $\Omega(n^{\log n})$ lower bounds on BD^k for parity^{c)}, but in doing this introduced important ideas which were subsequently valuable in proving the depth- k complexity of parity to be exponential. This final step in separating a relativised polynomial-time hierarchy was achieved by Yao (1985). Yao's proof is extremely complicated but Hastad (1986) discovered a simpler argument which gave improved exponential lower bounds. It is this proof which we present below in Section(5.3). Common to all three arguments is the employment of probabilistic counting techniques c.f Chapter(3), Section(3.5.1.3).

The fact that parity functions could not be realised by simultaneous polynomial size and constant depth networks motivated the investigation of several issues related to depth- k networks. In Section(5.4) we describe some results which allow further exponential

b) The same paper also constructs an oracle, B , for which $P^B = NP^B$. This raises difficulties in trying to establish $P \neq NP$ from relativisation results. A fuller discussion of these problems may be found in (Hopcroft and Ullman, 1979).

c) Tkachev (1980) independently proved superpolynomial lower bounds for parity when realised by Σ_3 networks.

lower bounds on depth- k complexity to be deduced. These are of two kinds: those obtained via constant-depth reductions, as first outlined in (Furst et al., 1984); and those derived as a consequence of the main lemma proved in Hastad (1986).

The final section of this chapter deals with a different bounded-depth model: Section(5.5) describes some recent results of Razborov (1986) on the complexity of $\{\wedge, \oplus\}$ depth- k networks.

Before these the work of Lupanov on universal depth- k formulae is presented in Section(5.2).

5.2) Universal bounds on bounded-depth formulae

Lupanov (1961a) introduced bounded-depth networks as a generalisation of DNF and CNF. In this section we prove asymptotically matching upper and lower bounds on the number of 2-input \wedge and \vee gates required to compute any Boolean function by depth- k formulae. Note that by considering only constant fan-in gates, the concept of depth- k formula becomes the restriction of permitting only k alternating levels of gate operations, e.g depth-3 formulae with arbitrary fan-in are equivalent to formulae over the basis $\{\wedge, \vee\}$ having negated inputs, in which every path from an input node to the output gate consists of a sequence of \wedge -gates, followed by a sequence of \vee -gates, followed by a sequence of \wedge -gates. We call such networks k -alternation or bounded alternation formulae. There is a close relationship between this measure and the number of wires in a depth- k network.

Fact 5.1: For a bounded alternation formula, T , let $\mathbf{L}^k(T)$ denote the number of 2-input gates as before. Furthermore for a depth- k formula, S , let $\mathbf{B}^k(S)$ denote the value of $BD^k(S)$ minus the number of (arbitrary fanin) gates in S . For $f \in B_n$ the measures $\mathbf{L}^k(f)$ and $\mathbf{B}^k(f)$ are defined in the obvious way.

For all $f \in B_n$

$$\mathbf{L}^k(f) = \mathbf{B}^k(f)$$

Proof: i) $\mathbf{L}^k(f) \leq \mathbf{B}^k(f)$: Let T be a depth- k formula realising f . We proceed by induction on the number of gates in T to construct a k -alternation formula, S , realising f and satisfying $\mathbf{L}^k(S) \leq \mathbf{B}^k(T)$. For the inductive base, T consists of a single gate with fan-in p say. The equivalent k -alternation formula contains $p-1$ two input gates, with the same operation as the single gate of T . Since the only wires in T are the p inputs this proves the inductive base. Now assume that the upper bound for all T containing fewer than t gates and let T be a depth- k formula realising f and containing exactly t gates. Without loss of generality let the output gate of T be an \wedge -gate and have fan-in $p \geq 2$. Then this gate computes the conjunction of p depth- $(k-1)$ formulae each containing at most $t-1$ gates. Let T_1, \dots, T_p denote these. By the inductive hypothesis there are $(k-1)$ -alternation formulae, S_1, \dots, S_p such that for each i , S_i computes the same function as T_i and,

$$\mathbf{L}^{k-1}(S_i) \leq \mathbf{B}^{k-1}(T_i)$$

Let S be the k -alternation formula which is formed by computing the conjunction of S_1, \dots, S_p using $p-1$ \wedge -gates. We have,

$$\begin{aligned} \mathbf{L}^k(S) &\leq p-1 + \sum_{i=1}^p \mathbf{L}^{k-1}(S_i) \\ &\leq p-1 + \sum_{i=1}^p \mathbf{B}^{k-1}(T_i) \\ &= \mathbf{B}^k(T) \end{aligned}$$

and this completes the inductive step and proof of the upper bound. The proof that $\mathbf{L}^k(f) \geq \mathbf{B}^k(f)$ is carried out in a similar manner using induction on the $\mathbf{L}^k(S)$ to construct a depth- k formula T computing f and having $\mathbf{B}^k(T) \leq \mathbf{L}^k(S)$. This is left to the reader. \square

The following upper bounds on formulae are the best possible for all $k \geq 2$; and we can prove exact bounds for the case $k=2$. The lower bound proved for this last result will be important in Section(5.2). We employ the notation,

$$\mathbf{L}^k(B_n) = \max \{ \mathbf{L}^k(f) : f \in B_n \}$$

Theorem 5.1: (Lupanov, 1961a)

$$\mathbf{L}^2(B_n) = n 2^{n-1} - 1$$

Proof: For the upper bound observe that any $f \in B_n$ has either at most 2^{n-1} satisfying assignments or at most 2^{n-1} unsatisfactory assignments. Suppose it is the former. Expressing f in DNF yields a formula of size at most $2^{n-1}(n-1) + 2^{n-1} - 1$ and this gives the upper bound. An identical argument applies if there are fewer than 2^{n-1} satisfying assignments by using a CNF representation of f .

For the lower bound consider either of the parity functions, i.e $\bigoplus_{i=1}^n x_i$ or its negation. Since the number of gates in a 2-alternation formula realising f is just the number of gates used to compute a DNF representation of f (for the ordering \wedge followed by \vee) or a CNF representation (for the alternative ordering), to prove the lower bound it suffices to show that any DNF (resp. CNF) representation of a parity function must have at least 2^{n-1} implicants (resp. clauses) and each implicant (clause) depends on all n variables. We give the proof for DNF. The other case is proved identically. Let $f(\mathbf{X}_n)$ be a parity

function and consider any DNF representing f . Each product term in this must depend on all n variables of \mathbf{X}_n , for if there is a product p , such that $p \leq f$ and p does not depend on x_i , then the assignments α and β which make exactly $\text{var}(p)$ and $\text{var}(p) \cup \{x_i\}$ take the value 1, both satisfy f . But this would contradict the definition of parity function. It now follows that since all implicants of f depend on all variables there must be exactly 2^{n-1} product terms in any DNF expressing f . This is because any such implicant is satisfied by exactly one assignment and there are 2^{n-1} satisfying assignments for any parity function. \square

Theorem 5.2: (Lupanov, 1961a; 1973) For all $k \geq 3$, $\mathbf{L}^k(B_n) \sim 2^n / \log n$.

Proof: This result is stated (without a detailed proof) in Lupanov (1961a); the presentation below follows Lupanov (1973). The lower bound, which holds for almost all Boolean functions, is immediate from Theorem(4.1) since the k -alternation restriction involves a subset of all formulae. It is clear that $\mathbf{L}^{k+1}(f) \leq \mathbf{L}^k(f)$ so it suffices to prove the upper bound for $k=3$. For this we return to the expansion of $f \in B_n$ used in proving the optimal upper bound on network depth in Theorem(2.11). Recall that this partitions \mathbf{X}_n into 4 sets \mathbf{W} , \mathbf{Y} , \mathbf{Z} and \mathbf{U} of sizes w , y , z and u , with which $f(\mathbf{W}, \mathbf{Y}, \mathbf{Z}, \mathbf{U})$ is

$$\bigvee_{\sigma} \bigvee_i \bigvee_k \bigvee_j \delta_{\sigma}(\mathbf{W}) \phi_i(\mathbf{U}) g_{i,\sigma,k,j}(\mathbf{Y}, \mathbf{Z}) \bigwedge_{\rho} (\neg \delta_{\rho}(\mathbf{Y}) \vee f_{i,\sigma,\rho,k,j}^{(3)}(\mathbf{U}))$$

In this, σ ranges over all assignments to \mathbf{W} ; $1 \leq i \leq 2^u/u$; $1 \leq k \leq \lceil 2^z/s \rceil$; ρ ranges over all assignments to \mathbf{Y} ; $1 \leq j \leq N$ and $N \leq \frac{u}{q} + 2^s$. Here u is a power of 2, $q \leq u$ and s, q are parameters to be fixed subsequently. Further recall that $\phi_i(\mathbf{U})$ is the characteristic function of the sphere with centre $\alpha^{(i)} \in \{0, 1\}^u$; that $g_{i,\sigma,k,j}$ is some function of \mathbf{Y} and \mathbf{Z} ; and $f_{i,\sigma,\rho,k,j}^{(3)}$ is the disjunction of at most q

variables from \mathbf{U} .

To simplify presentation of the upper bound we will adopt the following notations:

$$F_{i,\sigma,\rho,k,j}^1 = f_{i,\sigma,\rho,k,j}^{(3)}(\mathbf{U}) \quad (5.1)$$

$$F_\rho^2 = \neg \delta_\rho(\mathbf{Y}) \quad (5.2)$$

$$F_{i,\sigma,\rho,k,j}^3 = F_{i,\sigma,\rho,k,j}^1 \vee F_\rho^2 \quad (5.3)$$

$$F_{i,\sigma,k,j}^4 = \bigwedge_\rho F_{i,\sigma,\rho,k,j}^3 \quad (5.4)$$

$$F_{i,\sigma,k,j}^5 = g_{i,\sigma,k,j}(\mathbf{Y}, \mathbf{Z}) \quad (5.5)$$

$$F_i^6 = \phi_i(\mathbf{U}) \quad (5.6)$$

$$F_\sigma^7 = \delta_\sigma(\mathbf{W}) \quad (5.7)$$

$$F_{i,\sigma,k,j}^8 = F_{i,\sigma,k,j}^4 \wedge F_{i,\sigma,k,j}^5 \wedge F_i^6 \wedge F_\sigma^7 \quad (5.8)$$

$$F^9 = \bigvee_i \bigvee_\sigma \bigvee_k \bigvee_j F_{i,\sigma,k,j}^8 \quad (5.9)$$

F^9 , i.e. $f(\mathbf{W}, \mathbf{Y}, \mathbf{Z}, \mathbf{U})$ is realised as a formula (using 2-input gates) from the class Σ_3 , thus the alternating levels of gates are $\vee \wedge \vee$. We first show that the constructed definition of F^9 is a formula in this class. The functions F^r for $1 \leq r \leq 9$ are realised by corresponding formulae G^r in the following classes.

$$\left[\begin{array}{l} G^1 \in \Sigma_1 \\ G^2 \in \Sigma_1 \\ G^3 \in \Sigma_1 \\ G^4 \in \Pi_2 \\ G^5 \in \Pi_2 \\ G^6 \in \Pi_2 \\ G^7 \in \Pi_1 \\ G^8 \in \Pi_2 \\ G^9 \in \Sigma_3 \end{array} \right]$$

Using \mathbf{L}_Σ^k and \mathbf{L}_Π^k to denote the number of 2-input gates in a Σ_k or Π_k formula, it is easy to see that:

$$\left[\begin{array}{l} \mathbf{L}_\Sigma^1(G^1) \leq q \\ \mathbf{L}_\Sigma^1(G^2) \leq y \\ \mathbf{L}_\Sigma^1(G^3) \leq y + q \\ \mathbf{L}_\Pi^2(G^4) \leq 2^y (y + q) \end{array} \right]$$

For G^5 , since $F_{i,\sigma,k,j}$ is a function of $y+z$ variables its CNF (i.e Π_2 representation) is of size at most $2^{y+z} (y+z)$. For G^6 , which is computed as the CNF of $\phi_i(\mathbf{U})$ we claim that $\mathbf{L}_\Pi^2(G^6) \leq u^2$. To see this recall that ϕ is the characteristic function of a sphere with centre $\alpha = \langle a_1, \dots, a_u \rangle \in \{0, 1\}^u$. Thus the prime implicants of $\phi(\mathbf{U})$ are the u products,

$$\bigcup_{i=1}^u \{ u_1^{a_1} \dots u_{i-1}^{a_{i-1}} u_i^{-a_i} u_{i+1}^{a_{i+1}} \dots u_u^{a_u} \}$$

An assignment to \mathbf{U} satisfies $\phi(\mathbf{U})$ if and only if it differs in exactly one place from the centre α . It follows that

$$\phi(\mathbf{U}) = T_1^u(u_1^{-a_1}, \dots, u_u^{-a_u}) \wedge \neg T_2^u(u_1^{-a_1}, \dots, u_u^{-a_u})$$

and this is,

$$\left(\bigvee_{i=1}^u u_i^{\neg a_i} \right) \wedge \bigwedge_{1 \leq i < j \leq u} (u_i^{\neg a_i} \vee u_j^{\neg a_j})$$

which is easily verified as having size u^2 .

G^7 clearly satisfies $\mathbf{L}_{\Pi}^1(G^7) \leq w$ and

$$\mathbf{L}_{\Pi}^2(G^8) \leq 2^y (q + y + 2^z (y + z)) + u^2 + w$$

From this we have,

$$\begin{aligned} \mathbf{L}_{\Sigma}^3(G^9) &\leq \frac{2^u}{u} 2^w p N (2^y (q + y + 2^z (y + z)) + u^2 + w) \\ &\leq \frac{2^{u+w}}{u} \left(\frac{2^z}{s} + 1 \right) \left(\frac{u}{q} + 2^s \right) (2^y (q + y + 2^z (y + z)) + u^2 + w) \end{aligned}$$

If we set $y = \lfloor 2 \log n \rfloor$, $z = \lfloor 2 \log \log n \rfloor$, $u = 2^{\lfloor \log n \rfloor - 1}$, $q = \lfloor (\log n)^4 \rfloor$ and $s = \lfloor \log n - 5 \log \log n \rfloor$ then,

$$u = \frac{n}{2} ; y = o(q) ; 2^z (y + z) = o(q)$$

$$w + u^2 = o(q 2^y) ; s = o(2^z) ; q 2^s = o(u)$$

and so,

$$\mathbf{L}^3(f) \leq (1 + \varepsilon) \frac{2^n}{s} = (1 + \varepsilon) \frac{2^n}{\log n}$$

for all $\varepsilon > 0$ and sufficiently large n . This complete the proof of the upper bound. \square

5.3) Exponential Lower Bounds On Parity Functions

We know from the lower bound proved in Theorem(5.1) that the parity functions $\bigoplus_{i=1}^n x_i$ and $\neg \bigoplus_{i=1}^n x_i$ have exponential depth-2 complexity. In this section it is shown that for all constant k any depth- k formula realising a parity function of n variables must have size exponential in n . The proof below is that of Hastad (1986), which improves and simplifies the earlier proof of this result given in Yao (1985). The key idea in Hastad's argument is a result which shows that any "small" CNF-formula of n variables can be "simplified" to a "small" DNF-formula of m variables cf. Main Lemma below. This result is used to construct an inductive proof that parity functions require exponential size depth- k formulae: for some constant c , any depth- k networks realising n -input parity of size $< 2^{c n^{\frac{1}{k-1}}}$ can be used to prove the existence of depth-2 formulae realising parity functions of m variables but with size $< 2^m$.

As was mentioned in the introduction the proof relies on probabilistic counting techniques. In trying to prove that parity required exponential size bounded-depth formulae, one might proceed by noting that since \wedge (\vee) gates with variables as inputs can be eliminated by setting a chosen variable to 0 (1), a depth- k formula could be simplified by finding an appropriate choice of values for a subset of the inputs which would allow the initial 3 layers, $\wedge - \vee - \wedge$ say, to be transformed into 2 layers, $\vee - \wedge$, without greatly increasing the formula size. Observing that all sub-functions of parity functions are again parity functions then permits an inductive proof to be constructed i.e depth-2 formulae for parity have exponential complexity; the existence of sub-exponential depth- k formulae for parity implies the existence of sub-exponential depth- $k-1$ formulae for parity. Problems arise in trying to construct appropriate reducing assignments

explicitly, since the number of gates at the top level may be very large and assignments which eliminate most of these may result in the subsequent parity function being dependent on too few variables to make the induction work. Following the lead of Furst et al. (1984), Hastad proved the *existence* of appropriate assignment to input variables by using probabilistic methods.

Definition 5.2: Let $p \in (0, 1)$. R_p is the class of random partial assignments π for which,

$$\text{Prob} [x_i^{\pi} = 0] = \frac{1-p}{2}$$

$$\text{Prob} [x_i^{\pi} = 1] = \frac{1-p}{2}$$

$$\text{Prob} [x_i \notin \text{var}(\pi)] = p$$

these events occurring independently for each x_i . •

We will use $\pi(x_i) = 0$, $\pi(x_i) = 1$ and $\pi(x_i) = *$ to denote the three possible outcomes. Note that if $\pi \in R_p$ is applied to \mathbf{X}_n then the expected number of variables which are not set to constants is pn . Thus choosing a large value of p increases the expected number of variables remaining after simplifying a depth- k formula using $\pi \in R_p$ but also decreases the probability of being able to restructure to a depth- $k-1$ formula efficiently since fewer gates are likely to be eliminatable.

We can now state Hastad's Main Lemma.

Lemma 5.2 (Main Lemma): Let F be a CNF formula in which each clause contains at most t literals from \mathbf{X}_n . Let π be a random partial assignment in R_p . The probability that F^{π} cannot be expressed in DNF using implicants of fewer than s variables is no more than $\alpha_{p,t}^s$, $\alpha_{p,t}$ being the unique positive root of,

$$\left(1 + \frac{4p}{(1+p)\alpha_{p,t}}\right)^t - \left(1 + \frac{2p}{(1+p)\alpha_{p,t}}\right)^t - 1 = 0$$

(If $p = o(1)$ it is straightforward to show that

$$\alpha_{p,t} \sim \frac{2pt}{\log_e \phi} < 5pt$$

where ϕ is the Golden Ratio.)

This result is an easy corollary of the following Lemma, which although technically stronger is simpler to prove. Recall that a prime implicant of $f \in B_n$ is a product of literals m such that $m \leq f$ and no sub-product of m is an implicant of f . For a given CNF, F , let $\text{rank}(F)$ denote the number of literals in the longest prime implicant of the function represented by F .

Lemma 5.3: Let $F = \bigwedge_{i=1}^w F_i$, where each F_i is the disjunction of at most t literals from \mathbf{X}_n . Let $\pi \in R_p$ and g be an arbitrary Boolean function. Then,

$$\text{Prob} [\text{rank}(F^{|\pi|}) \geq s \mid g^{|\pi|} = 1] \leq \alpha_{p,t}^s$$

Proof: (Note that this lemma implies Lemma(5.2) simply by choosing $g = 1$.) The proof is by induction on $w \geq 0$. The inductive base, $w=0$, is immediate since $F=1$ and hence $\text{rank}(F)=0$. Assuming the lemma holds for all values $\leq w-1$ we show it holds for F a conjunction of w clauses. Consider the effect of π on F_1 , the first clause of F . Either $F_1^{|\pi|} = 1$ or $F_1^{|\pi|} \neq 1$. Hence,

$$\text{Prob} [\text{rank}(F^{|\pi|}) \geq s \mid g^{|\pi|} = 1]$$

is bounded above by,

$$\max \left\{ \begin{array}{l} \text{Prob} [\text{rank}(F^{|\pi}) \geq s \mid g^{|\pi} = 1 \wedge F_1^{|\pi} = 1] \\ \text{Prob} [\text{rank}(F^{|\pi}) \geq s \mid g^{|\pi} = 1 \wedge F_1^{|\pi} \neq 1] \end{array} \right. \quad (5.10)$$

To prove the lemma it suffices to show that the probability of either of these events occurring is at most $\alpha_{p,t}^s$. Consider the first term of (5.10). Only those π which render F_1 equal to 1 are relevant in bounding this, so the given probability is that of the $w-1$ clause $\bigwedge_{i=2}^w F_i$ having a prime implicant dependent on at least s variables, given that the function $(g \wedge F_1)$ becomes 1 when the partial assignment π is applied. The inductive hypothesis now yields the upper bound. Note that since the lemma is stated for all g , the fact that the conditional probability is based on $g \wedge F_1$ is catered for already.

Bounding the second term in (5.10) is rather more difficult. Let T be the set of literals upon which F_1 depends and without loss of generality assume that, $F_1 = \bigvee_{x_i \in T} x_i$, i.e no negated literals occur. We may assume this since the probability of setting $x_i := 0$, rendering the literal \bar{x}_i equal to 1, is identical to that of setting $x_i = 1$ which makes the literal x_i equal 1.

Any $\pi \in R_p$ may be viewed as the composition of two partial assignments: π_1 , which fixes only variables in T , and π_2 which sets other variables. Given this, the condition $F_1^{|\pi} \neq 1$ is equivalent to the condition $F_1^{|\pi_1} \neq 1$. Now the condition $F_1^{|\pi_1} \neq 1$ holding implies two facts:

- i) Some (non-empty) subset Y of T is left unaffected by π_1 .
- ii) Each prime implicant m of $F^{|\pi}$ contains at least one variable which occurs in Y .

Given $Y \subseteq T$ let $\mathbf{PI}_Y(F^{|\pi})$ denote the set of those prime implicants, m of $F^{|\pi}$, for which $Y = T \cap \text{var}(m)$. Furthermore, let $\text{rank}(F^{|\pi}, Y)$

denote the length of the longest prime implicant in $\mathbf{PI}_Y(F^{|\pi|})$ and $\pi_1(Y)=*$ the event " $\pi_1(x_i)=*$ for each $x_i \in Y$ ".

The second term of (5.10) is now at most,

$$\sum_{Y \subseteq T, Y \neq \emptyset} \text{Prob} [\text{rank}(F^{|\pi|}, Y) \geq s \mid g^{|\pi|} = 1 \wedge F_1^{|\pi_1|} \neq 1]$$

and this is no more than,

$$\sum_{Y \subseteq T, Y \neq \emptyset} E_1 E_2 \quad (5.11)$$

where,

$$E_1 = \text{Prob} [\pi_1(Y) = * \mid g^{|\pi|} = 1 \wedge F_1^{|\pi_1|} \neq 1]$$

$$E_2 = \text{Prob} [\text{rank}(F^{|\pi|}, Y) \geq s \mid g^{|\pi|} = 1 \wedge F_1^{|\pi_1|} \neq 1 \wedge \pi_1(Y) = *]$$

Now if,

$$E_1 \leq \left(\frac{2p}{1+p} \right)^{|Y|} \quad (5.12)$$

$$E_2 \leq (2^{|Y|} - 1) \alpha_{p,t}^{s-|Y|} \quad (5.13)$$

then it is easy to show that the expression of (5.11) is at most $\alpha_{p,t}^s$ and this will establish the lemma.

To see that (5.12) holds, first observe that for events A , B and C the inequality $\text{Prob} [A \mid B \wedge C] \leq \text{Prob} [A \mid C]$ holds if and only if the inequality $\text{Prob} [B \mid A \wedge C] \leq \text{Prob} [B \mid C]$ holds. So choosing A as the event " $\pi_1(Y) = *$ ", B as the event " $g^{|\pi|} = 1$ " and C as " $F_1^{|\pi_1|} \neq 1$ " it follows that (5.12) holds if both

$$\text{Prob} [A \mid C] \leq \left(\frac{2p}{1+p} \right)^{|Y|} \quad (5.14)$$

$$\text{Prob} [B | A \wedge C] \leq \text{Prob} [B | C] \quad (5.15)$$

hold.

(5.15) is obvious from our choice of A , B and C ; informally (5.15) asserts that forcing some variables of F_1 to be unaffected by π_1 cannot increase the probability of $g^{|\pi}$ being 1. For (5.14) the condition C , i.e. $F_1^{|\pi_1} \neq 1$ is equivalent to

$$\forall x \in T \pi_1(x) \in \{0, *\}$$

(recall that it is assumed that F_1 contains only positive literals). Thus,

$$\text{Prob} [\pi(x_i) = * | \pi(x_i) \in \{0, *\}] = \frac{\text{Prob} [\pi(x_i) = *]}{\text{Prob} [\pi(x_i) \in \{0, *\}]} = \frac{2p}{1+p}$$

and

$$\text{Prob} [\pi(x_i) = 0 | \pi(x_i) \in \{0, *\}] = \frac{\text{Prob} [\pi(x_i) = 0]}{\text{Prob} [\pi(x_i) \in \{0, *\}]} = \frac{1-p}{1+p}$$

(5.14) follows since these probabilities are independent.

Now consider the factor E_2 of (5.11). In estimating this only prime implicants $m \in \mathbf{PI}_Y(F^{|\pi})$ are relevant, where $F_1^{|\pi} \neq 1$. We may express any such m as $m_1 \wedge m_2$ where $\text{var}(m_1) = Y$ and $\text{var}(m_2) \subseteq \mathbf{X}_n - T$; this partition is possible from the fact that $m \in \mathbf{PI}_Y(F^{|\pi})$ and hence does not depend on any variable in $T - Y$. Now if σ is the partial assignment which fixes exactly the literals in m_1 to 1, then m_2 is a prime implicant of the function $F^{|\pi \sigma}$. So by considering the function $F^{|\pi \sigma}$ instead of $F^{|\pi}$ and $g^{|(\pi_1 \sigma)|\pi_2}$ instead of $g^{|\pi}$ we could employ the inductive hypothesis, provided that the condition $F_1^{|\pi_1} \neq 1$ could be removed. To accomplish this we maximise over all π_1 for which,

$$\pi_1(Y) = * \text{ and } \pi_1(T) \in \{0, *\}^{|T|}$$

Noting that $\text{rank}(F^{|\pi|}, Y) \geq s$ implies, from the definition of σ , that $\text{rank}(F^{|\pi^\sigma|}) \geq s - |Y|$, this gives,

$$E_2 \leq \sum_{\sigma \in \{0,1\}^{|Y|} - \{0\}^{|Y|}} \left(\max_{\pi_1(Y)=*, \pi_1(T) \in \{0,*\}^{|T|}} E_3 \right) \quad (5.16)$$

where E_3 is the probability of an appropriate π_2 being selected, i.e

$$E_3 = \text{Prob} [\text{rank}(F^{|\pi_1^\sigma \pi_2|}) \geq s - |Y| \mid g^{|\pi_1^\sigma \pi_2|} = 1]$$

Since $F^{|\pi_1^\sigma}$ is a conjunction of $w - 1$ clauses, applying the inductive hypothesis allows the conclusion $E_3 \leq \alpha_{p,t}^{s-|Y|}$. There are at most $2^{|Y|} - 1$ terms in the summation (5.16) so certainly,

$$E_2 \leq (2^{|Y|} - 1) \alpha_{p,t}^{s-|Y|}$$

This completes the proof that (5.13) holds and the inductive step. \square

It should be clear that this lemma also holds in a dual form for converting DNF formulae to CNF.

For a depth- k formula, T , the *bottom fan-in* of T is defined to be the maximal fan-in of any gate at depth 1 in T . An exponential lower bound on the size of depth- k formulae realising parity functions is easily deduced from the following lemma.

Lemma 5.4: Let $\beta(n, k) = 0.1 n^{\frac{1}{k-1}}$. There exists a constant n_0 such that for all $k \geq 2$ and $n \geq n_0^{k-1}$, n -input parity functions cannot be computed by depth- k formulae having bottom fan-in t and containing at most 2^s gates of depth at least 2, where $t \leq \beta(n, k)$ and $s \leq \beta(n, k)$.

Proof: By induction on $k \geq 2$. The inductive base has already been established in Theorem(5.1), which established that depth-2 formulae

for parity functions must have bottom fan-in n . So assume the lemma holds for depths $\leq k - 1$ and suppose that F is a depth- k formula realising a parity function on $n \geq n_0^{k-1}$ variables but having bottom fan-in $t \leq \beta(n, k)$ and fewer than 2^s gates of depth at least 2, for $s \leq \beta(n, k)$. Without loss of generality it may be assumed that the gates at depth 2 in F are all \wedge -gates. Let F_i be the sub-formula of F represented by the i 'th \wedge -gate at depth-2 in F , where $1 \leq i \leq 2^s$. F_i has bottom fan-in $\leq t$. From the Main Lemma, using $p = 0.1 \beta(n, k)^{-1}$, $s = t = \beta(n, k)$, the probability that a random partial assignment $\pi \in R_p$ leaves $\text{rank}(F_i^{|\pi|}) \geq s$ is at most $\alpha_{p,t}^s$. Hence the probability that π leaves some F_i having $\text{rank}(F_i^{|\pi|}) \geq s$ is at most $2^s \alpha_{p,t}^s \leq (2 \alpha_{p,t})^s$. In addition for n large enough the probability that $F^{|\pi|}$ depends on at least

$$m = n p = n^{\frac{k-2}{k-1}}$$

variables is at least $1/3$. It follows that the probability that $F^{|\pi|}$ depends on fewer than m variables or that there is some F_i for which $\text{rank}(F_i^{|\pi|}) \geq s$ is at most

$$\frac{2}{3} + (2 \alpha_{p,t})^s$$

Since $\alpha_{p,t} < 1/2$, for large enough n , this probability is less than 1. It follows that there certainly exists a partial assignment π with which $F^{|\pi|}$ depends on at least m variables and which allows each $F_i^{|\pi|}$ to be re-written as a DNF formula having bottom fan-in no more than s . Suppose such a π is applied to F and the formula $F^{|\pi|}$ re-written so that each $F_i^{|\pi|}$ is expressed as a DNF formula; let G be the resulting formula which computes a parity function of at least m variables, has bottom fan-in at most s and depth $k - 1$ since there are two adjacent \vee -levels in G which may be collapsed to a single level (i.e levels 2 and 3). Note that the number of gates of depth at least 2 in G is still

at most 2^s .

Now

$$\begin{aligned} s \leq \beta(n, k) &= 0.1 n^{\frac{1}{k-1}} \\ &= 0.1 (n^{\frac{k-2}{k-1}})^{\frac{1}{k-2}} \\ &\leq 0.1 m^{\frac{1}{k-2}} = \beta(m, k-1) \end{aligned}$$

Also,

$$m \geq n^{\frac{k-2}{k-1}} \geq (n_0^{k-1})^{\frac{k-2}{k-1}} = n_0^{k-2}$$

These contradict the inductive hypothesis and so F does not exist. \square

Theorem 5.3: Let $\gamma(n, k) = 2^{0.1(0.3n)^{\frac{1}{k-1}}}$. There exists a constant n_0 , such that for all $k \geq 2$ any depth- k formula computing a parity function of $n \geq n_0^k$ variables, contains at least $\gamma(n, k)$ gates.

Proof: Suppose the theorem does not hold, so that there is a depth- k formula realising a parity function of n variables containing fewer than $\gamma(n, k)$ gates. Such a formula may be regarded as one of depth $k+1$ having bottom fan-in 1. Let $p=0.3$, $t=1$ and $s=\log \gamma(n, k)$. Using arguments similar to the proof of Lemma(5.4) we find a partial assignment π which leads to a depth- $k-1$ formula realising parity of $m \geq 0.3n$ variables which has bottom fan-in $s \leq \beta(m, k-1)$ and fewer than $2^{\beta(m, k-1)}$ gates of depth at least 2. But this contradicts Lemma(5.4) which showed that such formulae do not exist. \square

5.4) Consequences of the Parity Function Lower Bound

The work commenced in (Furst et al., 1984) and its subsequent development by Yao (1985) and Hastad (1986) gives rise to a number of further questions concerning bounded-depth networks, some of which we examine in this section.

(Furst et al., 1984) showed that the depth- k complexity of several natural Boolean functions was polynomially related to the depth- k complexity of parity. These results rely on a concept of *constant-depth reducibility* which was formalised and considered explicitly in (Chandra et al., 1984). This paper introduced the complexity class $S - D(S(n), D(n))$. A family, $[f_n]$ of n -input Boolean functions is in this class if and only if,

$$\forall n \geq 1, f_n \text{ is computable by a depth-}k \text{ network of size at most } S(n), \text{ for some (not necessarily constant) } k \leq D(n)$$

Specific cases of interest are

$$S - D(\text{poly}, \text{const}) = \bigcup_{c, d, k \geq 0} S - D(cn^k, d) \quad (5.17)$$

$$S - D(\text{poly}, D(n)) = \bigcup_{c, k \geq 0} S - D(cn^k, D(n)) \quad (5.18)$$

$$S - D(\text{poly}, \text{poly} - \log) = \bigcup_{c, k, d, l \geq 0} S - D(cn^k, d(\log n)^l) \quad (5.19)$$

We will also refer to the classes $POLY - \Sigma_k$ ($POLY - \Pi_k$) of families of Boolean functions which can be realised by polynomial size Σ_k (Π_k) networks; $POLY - \Sigma_k^m$ and $POLY - \Pi_k^m$ refer to the monotone variants of these classes.

Theorem (5.3) established that $PAR_n \notin S - D(\text{poly}, \text{const})$.

To investigate the structure of these classes (Chandra et al., 1984) considered two concepts of reducibility: reductions via p -projections, as defined by (Skyum and Valiant, 1985) viz Defn (2.1) above; and a weaker form known as *constant-depth truth-table* reducibility. Given two families $F = [f_n]$ and $G = [g_n]$ F is said to be constant-depth truth-table reducible to G , denoted $F \leq_{cd-tt} G$ if and only if there is a polynomial, $p(n)$ and a constant c such that,

$\forall n \geq 1$, f_n can be computed by a G -network of size $\leq p(n)$ and depth $\leq c$.

Here a G -network is defined similarly to a depth- k network but additionally permits gates which compute functions $g_j \in G$ provided that $j \leq p(n)$ and there are no paths from the outputs of any such gates to the inputs of other G gates.

For families F and G we use the notation $F \leq_{proj} G$ if F is reducible to G via a p -projection.

From the definitions of \leq_{cd-tt} and \leq_{proj} it is easy to verify the following lemma.

Lemma 5.5:

- i) \leq_{cd-tt} and \leq_{proj} are both reflexive and transitive relations.
- ii) $F \leq_{proj} G \iff F \leq_{cd-tt} G$.
- iii) Let $S(n)$ and $D(n)$ be monotone non-decreasing functions. If $F \leq_{cd-tt} G$ or $F \leq_{proj} G$ and $G \in S - D(S(n), D(n))$ then there is a polynomial, $p(n)$, and a constant c for which,

$$F \in S - D(p(n)S(p(n)), cD(p(n)))$$

Thus if $G \in S - D(poly, const)$ then $F \in S - D(poly, const)$ also. \square

As examples of such reductions we use the following functions, in addition to parity. A number of other examples are given in

(Chandra et al., 1984).

ADD and *MULT* are the functions which compute the sum (product) of two n -digit binary numbers. *COMP* takes as input two n -digit binary numbers, \underline{x} and \underline{y} , returning the result 1 if and only if $\underline{x} > \underline{y}$. *TC* takes as input n^2 Boolean variables encoding an adjacency matrix, $A = [a_{i,j}]$ and outputs the n^2 Boolean entries of the matrix A^+ being the transitive closure of A .

First some examples of efficiently computable functions are presented.

Theorem 5.4:

- i) $ADD \in S - D(\text{poly}, \text{const})$;
- ii) $COMP \in S - D(\text{poly}, \text{const})$.

Proof: i) Let $\underline{x} = x_{n-1} x_{n-2} \cdots x_0$ and $\underline{y} = y_{n-1} y_{n-2} \cdots y_0$ be the n -bit binary representations of the two numbers being added. The so-called carry look-ahead scheme is used. This proceeds by computing $f_i = x_i y_i$ and $g_i = x_i \oplus y_i = x_i \bar{y}_i \vee \bar{x}_i y_i$ for each i ($0 \leq i < n$). All of these can be computed in depth 2 using only $O(n)$ wires. The final stage is to compute each of the output sum bits, s_j $0 \leq j \leq n$; for this a sequence of carry bits, c_j , $0 \leq j < n$ must be computed. The carry bits are computed by

$$c_j = \bigvee_{i=0}^j g_i \wedge \bigwedge_{k=i+1}^j h_k$$

The sum bits are given by $s_0 = h_0$, $s_n = c_{n-1}$ and for $0 < j < n$ $s_j = h_j \oplus c_{j-1}$. The resulting network, with negation restricted to network inputs, clearly has polynomial size and constant-depth.

ii) $\underline{x} > \underline{y}$ if and only if there is some i , $0 \leq i < n$ for which $x_i = 1 > 0 = y_i$ and $x_j = y_j$ for each $i + 1 \leq j < n$. Whether this property holds can be tested by implementing the expression,

$$\bigvee_{i=0}^{n-1} (x_i \wedge \bar{y}_i \wedge \bigwedge_{j=i+1}^{n-1} (x_j \equiv y_j))$$

Since $x \equiv y = x y \vee \bar{x} \bar{y}$ this can be computed by a network of size $O(n^2)$ and depth 4. \square

Theorem 5.5:

- i) $PARITY \leq_{proj} MULT$
- ii) $PARITY \leq_{cd-tt} TC$

Proof:

i) Let \mathbf{X}_n be the input variables for an instance of a parity function and $r = \log_2 n$. Construct the two n -bit numbers;

$$P = \sum_{i=1}^n x_i 2^r ; Q = \sum_{i=1}^n 2^r$$

With these $PQ = \sum_{i=1}^n c_i 2^r$, where the c_i are r -bit numbers. The least significant bit of c_n gives the parity of $\sum_{i=1}^n x_i$ and this yields the result.

ii) As before let \mathbf{X}_n be the input variables for an instance of a parity function. Consider the following $n+2$ -vertex undirected graph, G . G has vertices $v_0, v_1, \dots, v_n, v_{n+1}$. There is an edge between v_0 and the lowest indexed v_i for which $x_i = 1$ and $x_j = 0, \forall j < i$. Similarly there is an edge between v_{n+1} and the highest indexed v_i for which $x_i = 1$ and $x_j = 0, \forall j > i$. Finally there are edges all pairs v_i and v_j such that $j > i, x_i = x_j = 1$ and $x_k = 0$ for all $i < k < j$. Let $A = [a_{i,j}]$, where $0 \leq i, j \leq n+1$ be the adjacency matrix corresponding to G . The entries of A are easily computed by Π_1 networks by using the identities,

$$\begin{aligned}
 a_{0,i} &= \left(\bigwedge_{j=1}^{i-1} \bar{x}_j \right) \wedge x_i \\
 a_{i,n+1} &= x_i \wedge \left(\bigwedge_{j=i+1}^n \bar{x}_j \right) \\
 a_{i,j} &= x_i \wedge \left(\bigwedge_{k=i+1}^{j-1} \bar{x}_k \right) \wedge x_j
 \end{aligned}$$

Now let $B=[b_{i,j}]$ be the adjacency matrix in which $b_{i,j}=1$ if and only if there is a path of containing exactly 2 edges between v_i and v_j in G . The entries of this matrix can be computed using Σ_2 networks, whose inputs are the $a_{i,j}$ computed previously, and the relation

$$b_{i,j} = \bigvee_{k=1}^n a_{i,k} a_{k,j}$$

Since all the $a_{i,j}$ are computed by Π_1 networks, it follows that all the $b_{i,j}$ are computed by Σ_2 networks. The final stage of the construction is to compute the transitive closure of B using a single transitive closure gate. Since $b_{i,j}=1$ if and only if there is a path of exactly 2 edges between i and j , it follows that $b_{i,j}^+=1$ if and only if there is a path containing an even number of edges between i and j . Any such path contains an odd number of vertices and so in the resulting matrix, $b_{0,n+1}^+=1$ if and only if $\sum_{i=1}^n x_i \equiv 1 \pmod{2}$. This completes the reduction. \square

The inductive proof of Theorem (5.3) requires only 2 properties of parity functions; that any subfunction of a parity function is again a parity function; that this function requires many (in fact all) of its inputs to be determined before its result is known. These properties are shared by other Boolean functions and it turns out that the techniques used in proving parity to be difficult can be applied almost directly in such cases.

For the class S_n of symmetric Boolean functions, (Fagin et al., 1985) used the results of (Furst et al., 1984) to characterise those symmetric functions having superpolynomial depth- k complexity. By applying Hastad's techniques Moran (1987), and independently (Brustmann and Wegener, 1986), generalised these results on symmetric functions.

Following Moran (1987) we introduce the terminology below.

Definition 5.3: Let $f \in S_n$ and $w_0 \cdots w_n$ be the spectrum of f . For $0 \leq j \leq n$ we say that j is a *left (right) boundary* of f if $w_j = 1$ and $w_{j-1} = 0$ ($w_{j+1} = 0$). j is a *boundary* if it is a left or a right boundary of f . $B(f) \subseteq \{0, 1, \dots, n\}$ denote the set of boundaries of f and $b(f)$ the value of $\max_{j \in B(f)} \min\{j, n-j\}$. •

(Fagin et al., 1985) proved,

Theorem 5.6: Let $p(n)$ be a polynomial, k a natural number and $\varepsilon > 0$ some constant. If $f \in S_n$ has a boundary j such that $n^\varepsilon \leq j \leq n - n^\varepsilon$ then $f \notin S - D(p(n), k)$. □

Moran (1987) extends the interval of this theorem to,

$$[(\log n)^r, n - (\log n)^r] \quad (5.20)$$

where r is any function of n such that $r \rightarrow \infty$.

In combination with (Fagin et al., 1985), which established that $f \in S - D(\text{poly}, \text{const})$ if $f \in S_n$ does not have a boundary in the interval given by (5.20), Moran's result completely characterises those symmetric functions which are not in $S - D(\text{poly}, \text{const})$ and also gives explicit lower bounds on f in terms of $b(f)$.

The results are obtained in two stages. First a lower bound on the size of depth- k networks computing functions with boundaries at $n/2$ is proved. This mirrors the proof of Lemma(5.4) and

Theorem(5.3). The second stage reduces arbitrary symmetric functions, of n variables, to ones of $n - m$ variables having boundaries at $(n - m)/2$. Prior to these, two simple technical lemmas are needed.

Lemma 5.6: Let $f \in S_n$ and π any partial assignment which fixes l variables to 1 and $n - m - l$ variables to 0. Then for all $0 \leq j \leq m - 1$, resp. $1 \leq j \leq m$ $f^{|\pi|}$ j is a right (left) boundary of $f^{|\pi|}$ if and only if $m + j$ is a right (left) boundary of f .

Proof: Let $w_0 \cdots w_n$ be the spectrum of f . For any assignment π as in the Lemma statement, the spectrum of $f^{|\pi|}$ consists of the subword $w_l w_{l+1} \cdots w_{l+m} = v_0 \cdots v_m$. The lemma now follows since j is a right (left) boundary of the function, $f^{|\pi|}$ with spectrum $v_0 \cdots v_m$ if and only if $l + j$ is a right (left) boundary of a function, f , whose spectrum contains the subword $w_l \cdots w_{l+m}$. \square

Lemma 5.7: Let $f \in S_n$. Any depth-2 network realising f has bottom fan-in at least $b(f)$.

Proof: Let T be a depth-2 network computing f and $j = b(f)$. First suppose that T is a Σ_2 -network. If j is a left boundary then there is some \wedge -gate, g , of T becoming 1 under the assignment, π , which fixes $x_i = 1$ for each $1 \leq i \leq j$ and all remaining variables to 0. Since j is a *left* boundary each x_i with $1 \leq i \leq j$ must be an input of g , for if not then modifying π so that x_i becomes 0, leaves the output of g unchanged. Hence if j is a left boundary then T has bottom fan-in at least j . A similar argument, identifying a gate with fan-in at least $n - j$, holds if j is a right boundary by fixing an additional variable to 1, i.e \bar{x}_i must be an input of g for each $j + 1 \leq i \leq n$.

Now suppose that T is a Π_2 -network. Consider the assignment, π , which fixes $x_i = 1$ for $1 \leq i \leq j - 1$ and all remaining variables to 0. If j is a left boundary then some \vee -gate, h of T is made 0 under π . The literal x_i must be an input of h , for each $j \leq i \leq n$ otherwise

increasing x_i to 1 leaves the result of h and T unchanged. If j is a right boundary we can identify an \vee -gate with fan-in at least $j+1$ in a similar manner by considering the assignment which fixes $x_i=1$ for $1 \leq i \leq j+1$.

In summary, if j is a left boundary then T has bottom fan-in at least $\min\{j, n-j+1\}$ if j is a right boundary then T has bottom fan-in at least $\min\{n-j, j+1\}$ hence T has bottom fan-in at least $\min\{j, n-j\}$. \square

Lemma 5.8: Let $f \in S_n$ have a boundary at $j=n/2$ and $\beta(n, k)$ be as in Lemma(5.4). There is a constant n_1 such that for all constant $k \geq 2$ and $n \geq n_1^{k-1}$, f cannot be computed by depth- k networks having bottom fan-in t and containing at most 2^s gates of depth at least 2, where $t \leq \beta(n, k)$ and $s \leq \beta(n, k)$.

Proof: By induction on $k \geq 2$. The inductive base is immediate from Lemma(5.7). The inductive step is identical to the proof of the same stage in Lemma(5.4), noting that with $p=0.1 \beta(n, k)^{-1}$ the probability that a random partial assignment $\pi \in R_p$ sets equal numbers of variables to 0 and 1 and leaves $m \geq np$ variables unassigned is at least $1/n$. With such an assignment $f|^\pi \in S_m$ has, from Lemma(5.6), a boundary at $m/2$, and so the inductive argument used in Lemma(5.4) can be applied directly. \square

Theorem 5.7: Let f be as Lemma(5.8). For all constant $k \geq 2$ any depth- k network computing f contains $\Omega(\gamma(n, k))$ gates, where $\gamma(n, k)$ is defined exactly as in Theorem(5.3).

Proof: Exactly as Theorem(5.3). \square

Theorem 5.8: Let $f \in S_n$, $b = b(f)/n$ (so that $0 < b \leq 0.5$) and $\eta(n, k, b) = 2^{0.1(0.6bn)^{\frac{1}{k-1}}}$. Every depth- k network realising f contains $\Omega(\eta(n, k, b))$ gates.

Proof: Without loss of generality let bn be a boundary of f , the case where $n - bn$ is a boundary of f is dealt with similarly. Let π be the partial assignment which sets $x_i = 0$ for each $1 \leq i \leq n - 2bn$ and leaves all other variables unset. $f^{|\pi} \in S_{2bn}$ and from Lemma(5.6) bn is a boundary of $f^{|\pi}$. These two facts and Theorem(5.7) now yield the theorem. \square

It should be noted that both Theorem(5.8) and Theorem(5.3) hold not only for constant k , but more generally for all $k \leq \log n / (\log \log n + C)$, for some constant C .

None of the preceding results indicate whether the sequence of complexity classes $[S - D(\text{poly}, k)]_{k=2}^{\infty}$ forms a proper hierarchy, i.e if

$$S - D(\text{poly}, k) \subset S - D(\text{poly}, k + 1) \quad \forall k \geq 2$$

The question was first resolved, affirmatively, by Sipser (1983). Sipser's proof is non-constructive, establishing the existence of families, $[f_n] \in S - D(\text{poly}, k)$ but not in $S - D(\text{poly}, k + 1)$. Hastad (1986) proved this result for specific families of functions. This is stated below without proof, as

Theorem 5.9: Let $n = m^k$ and

$$\mathbf{X}_n = \{ x_{i_1 i_2 \dots i_k} : 1 \leq i_1, i_2, \dots, i_k \leq m \}$$

Define,

$$F_{k,n}^{\vee}(\mathbf{X}_n) = \bigvee_{i_1} \bigwedge_{i_2} \bigvee_{i_3} \cdots \bigwedge_{i_j} \cdots \bigvee_{i_k} x_{i_1 \dots i_k}$$

where $Q_j \equiv \bigwedge$ if j is odd, and \bigvee if j is even.

$F_{k,n}^{\wedge}$ is defined similarly, but with $Q_j \equiv \bigvee$ if j is odd and \bigwedge if j is even.

For all constant $k \geq 3$;

$$F_{k,n}^{\vee} \in POLY - \Sigma_k ; F_{k,n}^{\vee} \notin S - D(poly, k - 1)$$

$$F_{k,n}^{\wedge} \in POLY - \Pi_k ; F_{k,n}^{\wedge} \notin S - D(poly, const) \quad \square$$

More generally the following containment results are known.

Theorem 5.10:

$$POLY - \Sigma_k \subset POLY - \Sigma_{k+1} \subset S - D(poly, const) \quad (5.21)$$

$$POLY - \Pi_k \subset POLY - \Pi_{k+1} \subset S - D(poly, const) \quad (5.22)$$

$$POLY - \Sigma_k \subset POLY - \Pi_{k+1} \quad (5.23)$$

$$POLY - \Pi_k \subset POLY - \Sigma_{k+1} \quad (5.24)$$

$$POLY - \Sigma_k^m \subset POLY - \Sigma_k \cap \{ [f_n] : f_n \in M_n \} \quad (5.25)$$

Proof: (5.21)-(5.24) are merely restatements of Theorem(5.9); (5.25) is from Okol'nishnikova (1982). \square

It is an open question as to whether $POLY - \Sigma_k = POLY - \Pi_k$, for $k \geq 3$, even in the monotone cases. That this does not hold for the case $k=2$ is easily shown by considering the functions $F_{2,n}^{\vee}$ and $F_{2,n}^{\wedge}$.

5.5) Bounded-Depth $\{\wedge, \oplus\}$ -formulae

The basis $\{\wedge, \vee, \neg\}$ has been shown to lack sufficient strength to compute efficiently a number of natural symmetric functions using depth- k circuitry. The results rely on the fact that the functions examined have exponentially many prime implicants and clauses thus neither short DNF nor CNF representations are possible. However in

Chapter(1) we described another normal form representation for Boolean functions: the ringsum expansion using the basis $\{\wedge, \oplus, 1\}$. Consider,

Definition 5.4: The class Λ_k of n -input depth- k formulae (over $\{\wedge, \oplus\}$) is inductively defined as follows:

i) If $k=0$ then,

$$\Lambda_k = \{x_1, \dots, x_n, 1 \oplus x_1, \dots, 1 \oplus x_n\}$$

ii) If $k>0$ and odd, then $S \in \Lambda_k$ if and only if S is the \oplus of some (possibly empty) set of formulae S_1, \dots, S_p , where $S_i \in \Lambda_{k-1}$.

iii) If $k>0$ and even, then $S \in \Lambda_k$ if and only if S is the \wedge of some (possibly empty) set of formulae S_1, \dots, S_p , where $S_i \in \Lambda_{k-1}$.

It will sometimes be convenient to regard $S \in \Lambda_k$ as the set of formulae $S_i \in \Lambda_{k-1}$ defining it and so write $S_{i-1} \in S$.

We use $BD_k^*(S)$ to denote the size (number of gates) in $S \in \Lambda_k$; for $f \in B_n$, $BD_k^*(f)$ is defined in the obvious way from this. •

There are two points which should be noted about this model. First the class Λ_3 does not correspond with the form of the ringsum expansion. The latter does not permit $x \oplus 1$ inside a product; clearly direct implementations of the ringsum expansion form a subset of Λ_3 . This contrasts with Σ_2 and Π_2 . Secondly $BD_k^*(f)$ is at most polynomially larger than $BD_{k+2}(f)$. This is immediate from the fact that,

$$\bigvee_{i=1}^n x_i = 1 \oplus \left(\bigwedge_{i=1}^n (1 \oplus x_i) \right)$$

cf De Morgan's Laws and the identity $\bar{x} = 1 \oplus x$.

It is obvious that parity functions, and hence families reducible to parity functions, have polynomial complexity in this model. Razborov

(1986) investigated the question of whether some simple function could be shown to require exponential size Λ_k formulae. This section presents the $\exp(\Omega(n^{1/k-2}))$ lower bound proved by him for the majority function. Our proof follows the simplified approach constructed by Paterson (1986) which yields a slightly improved lower bound. It is worth noting that the style of the proof is forced to be radically different from the arguments of Furst, Hastad et al., since \oplus can not be determined from a strict subset of its inputs and so there can be no analogue of, for example, Hastad's Main Lemma. Whereas results for $\{\wedge, \vee, \neg\}$ were obtained by largely combinatorial techniques, the lower bound proved by Razborov employs ideas from linear algebra, exploiting the correspondence between computation using $\{\wedge, \oplus, 1\}$ and formal polynomials over the field $\mathbf{GF}(2)$. The reader will observe some similarities to Razborov's methods for reasoning about monotone network complexity, given in Section(3.5.1); specifically the mapping of Λ_k formulae into a set-theoretic construct; and using the fact that this can be regarded as computing an approximation to the function considered.

Definition 5.5: For $f \in B_n$, let

$$|f| = |\{\alpha \in \{0, 1\}^n : f(\alpha) = 1\}|$$

For $H \subseteq B_n$ and $f \in B_n$ the *distance* of f from H is given by

$$\rho(f, H) = \min \{|f \oplus g| : g \in H\} \quad (5.26)$$

A *regular pattern* of depth- k , \mathbf{M} , is a sequence,

$$\mathbf{M} = \langle M_0, M_1, \dots, M_k; \Pi_1, \Pi_2, \dots, \Pi_k \rangle \quad (5.27)$$

where $M_i \subseteq B_n$, $\{x_i, 1 \oplus x_i : 1 \leq i \leq n\} \subseteq M_0$ and $\Pi_i : 2^{M_{i-1}} \rightarrow M_i$.

If $H \subseteq M_{i-1}$ then the *discrepancy* of H with respect to M_{i-1} , $\delta(H, i)$ is,

$$\delta(H, i) = |\Pi_i(H) \oplus_{f \in H}^* f| \tag{5.28}$$

where $*$ = \oplus if i is odd and \wedge otherwise. For a regular pattern, \mathbf{M} , as in (5.27) the *discrepancy of \mathbf{M}* , denoted $\Delta(\mathbf{M})$ is

$$\Delta(\mathbf{M}) = \max_{1 \leq i \leq k} \max_{H \subseteq M_{i-1}} \delta(H, i) \tag{5.29}$$

Finally the *outer cover* of \mathbf{M} is the set of functions comprising M_k . •

Comparing these with Definitions (3.12-13) the following lemma is analogous to Lemma(3.15).

Lemma 5.9: For all regular patterns, \mathbf{M} , of depth- k having outer cover M_k and for all $f \in B_n$,

$$BD_k^*(f) \geq \frac{\rho(f, M_k)}{\Delta(\mathbf{M})}$$

Proof: Let S be any Λ_k formula realising f . With each sub-formula, T , of S , where $T \in \Lambda_i$, we associate a function $f_T^M \in M_i$. Subsequently f_T will denote the function computed by a sub-formula T , so that $f_S = f$.

For $T \in \Lambda_i$, f_T^M is defined inductively as follows:

$$\begin{aligned} f_T^M &= f_T & \text{if } i = 0 \\ f_T^M &= \Pi_i(\{f_W^M : W \in T\}) & i > 0 \end{aligned} \tag{5.30}$$

We claim that for each T ,

$$|f_T^M \oplus f_T| \leq BD_k^*(T) \Delta(M) \tag{5.31}$$

f_T^M is the "approximation" to f_T when the computation by T is modelled by a computation using the regular pattern \mathbf{M} . (5.31) asserts that the number of points in $\{0, 1\}^n$ on which f_T and its approximation

differ can be bounded in terms of the discrepancy of the regular pattern \mathbf{M} , of the role of λ , δ_- and δ_+ in the proof of Lemma(3.15).

(5.31) is established by induction on i , for $T \in \Lambda_i$. The inductive base, $i=0$, is trivial since $f_T = f_T^M$ and so the LHS of (5.31) is equal to 0. Assuming (5.31) for all depths $< i$ we show it holds for i also. Clearly, the LHS of (5.31) is at most

$$\sum_{V \in T} |f_V^M \oplus f_V| + | \bigoplus_{V \in T}^* f_V^M \oplus \Pi_i(\{f_V^M : V \in T\}) |$$

i.e the total number of differences introduced in the sub-formulae of T plus the number of new differences introduced.

From the Inductive Hypothesis, (5.28) and (5.29) this is at most,

$$\sum_{V \in T} BD_{k-1}^*(V) \Delta(\mathbf{M}) + \Delta(\mathbf{M})$$

which is $BD_k^*(T) \Delta(\mathbf{M})$ as claimed.

The lemma now follows from (5.26) since $f_T^M \in M_k$ and so,

$$BD_k^*(f) \geq \frac{\rho(f, M_k)}{\Delta(\mathbf{M})} \quad \square$$

We now define a regular pattern of depth- k for which large lower bounds on distance and small upper bounds on discrepancy can be proved.

$P(d)$ denotes the linear space consisting of the set of formal polynomials in $\langle x_1, \dots, x_n \rangle$ over $\mathbf{GF}(2)$ having degree at most d . Any $g \in P(d)$ has the form,

$$g = c \oplus \bigoplus_{i=1}^l m_i$$

where m_i is a monom of size at most d and $c \in \{0, 1\}$. It is natural to

equate such polynomials with the Boolean functions they represent.

Recall the following two facts concerning this linear space:

$$\forall g, h \in P(d) \quad g \oplus h \in P(d) \quad (5.32)$$

$$\forall c \in \{0, 1\}, g \in P(d) \quad c \wedge g \in P(d) \quad (5.33)$$

$$\mathbf{M} = \langle M_0, \dots, M_k; \Pi_1, \dots, \Pi_k \rangle$$

is defined as follows.

For some parameter, r , to be fixed subsequently set

$$M_{2j} = M_{2j+1} = P(r^j) \quad \forall j \geq 0 \quad (5.34)$$

Π_i will be defined so that

$$\max_{H \subseteq M_{i-1}} \delta(H, i) \leq 2^{n-r} \quad (5.35)$$

When i is odd this is relatively easy; for $H \subseteq M_{i-1}$ simply set

$$\Pi_i(H) = \bigoplus_{f \in H} f \quad (5.36)$$

From (5.32) and (5.34) we have $\Pi_i(H) \in M_i$ and from (5.28) $\delta(H, i) = 0$.

That Π_i can be chosen to satisfy (5.35) when i is even is slightly harder to prove. This fact is established in,

Lemma 5.10: For any $H \subseteq P(d)$ there is some $g \in P(dr)$ such that $|\bigwedge_{f \in H} f \oplus g| \leq 2^{n-r}$.

Proof: (Paterson, 1986) For $H \subseteq P(d)$ let $h = \bigwedge_{f \in H} f$ and

$$H^* = \left\{ \bigoplus_{f \in H} e_f \cdot \neg f : e_f \in \{0, 1\} \right\}$$

so that H^* is the linear subspace spanned by $\{\neg f : f \in H\}$. Obviously for all $q \in H^*$ $q \wedge h = 0$.

Now let,

$$Null = \{ \alpha \in \{0, 1\}^n : h(\alpha) = 0 \}$$

We claim that $\forall \alpha \in Null$

$$|\{q \in H^* : q(\alpha) = 1\}| = |H^*|/2 \quad (5.37)$$

To see this note that,

$$h(\alpha) = 0 \iff \exists f \in H \text{ s.t. } f(\alpha) = 0$$

So for this f and any $q \in H^*$, $(q \oplus \neg f)(\alpha) \neq q(\alpha)$ and $q \oplus \neg f \in H^*$ since H^* is a linear space. (5.37) is immediate from these two facts.

From (5.37) it is clear that for each $S \subseteq Null$ we can find some $q \in H^*$ for which,

$$|S \cap \{\beta : q(\beta) = 1\}| \geq |S|/2 \quad (5.38)$$

Using (5.38) we can identify a sequence of functions q_1, \dots, q_r in H^* (for any $r \geq 1$), such that for $g' = \bigvee_{i=1}^r q_i$ it holds,

$$|\neg h \wedge g'| \geq (1 - 2^{-r}) |\neg h| \text{ and } g' \wedge h = 0 \quad (5.39)$$

This follows since obviously we can construct a sequence of pairwise disjoint sets, Def_i , $i = 1, 2, \dots, r$ for which

$$|Def_i| \geq |Null|/2^i$$

So from (5.38), for each $1 \leq i \leq r$, there is some $q_i \in H^*$ with which

$$|Def_i \cap \{\beta : q_i(\beta) = 1\}| \geq |Null|/2^{i+1}$$

Setting $g' = \bigvee_{i=1}^r q_i$, we have

$$\begin{aligned} |\neg h \wedge g'| &\geq \frac{1}{2} \sum_{i=1}^r \frac{|\neg h|}{2} \\ &= |\neg h|(1 - 2^{-r}) \end{aligned}$$

Also $h \wedge (\bigvee_{i=1}^r q_i) = \bigvee_{i=1}^r h q_i = 0$.

Lemma(5.10) follows by choosing $g = \neg g' = \bigwedge_{i=1}^r \neg q_i$. For with this choice, $g \in P(dr)$ and $h \leq g$, i.e

$$g(\alpha) = 0 \Leftrightarrow g'(\alpha) = 1 \Leftrightarrow h(\alpha) = 0$$

thus,

$$\begin{aligned} |h \oplus g| &= |\{\alpha \in \{0, 1\}^n : h(\alpha) = 0 \text{ and } g(\alpha) = 1\}| \\ &= |\neg h| - |\{\beta \in \{0, 1\}^n : \neg h(\beta) = 1 \text{ and } g'(\beta) = 1\}| \\ &\leq |\neg h| - |\neg h \wedge g'| \leq |\neg h|/2^r \\ &\leq 2^{n-r} \end{aligned}$$

since $|\neg h| = |Null| \leq 2^n$. \square

Lemma(5.10) shows that for $i = 2j$ we can define Π_i to satisfy $\delta(H, i) \leq 2^{n-r}$ for all $H \subseteq M_{i-1} = P(r^{j-1})$.

Corollary 5.1: For all $f \in B_n$ using the regular pattern, \mathbf{M} , of depth- k just defined

$$BD_k^*(f) \geq \frac{\rho(f, M_k)}{2^{n-r}}$$

Proof: Immediate from Lemmas(5.9) and (5.10). \square

In order to prove the existence of a symmetric Boolean function having large distance in \mathbf{M} , Razborov (1986) introduces a linear mapping, $R: B_n \rightarrow M_{A,B}$; here $M_{A,B}$ is the set of Boolean matrices whose rows are labelled with the elements of $A \subseteq \{0, 1\}^n$ and whose columns are labelled with the elements of $B \subseteq \{0, 1\}^n$. Paterson (1986) employs a simpler linear mapping which is used below. In what follows $\#_c(\alpha)$ denotes the number of positions in $\alpha \in \{0, 1\}^n$ which are equal to c ($c \in \{0, 1\}$) and for

$$\alpha = \langle a_1, \dots, a_n \rangle \in \{0, 1\}^n$$

γ_α is the monotone Boolean function,

$$\gamma_\alpha(\mathbf{X}_n) = \bigwedge_{i: a_i=1} x_i$$

Let $A \subseteq \{0, 1\}^n$, $B \subseteq \{0, 1\}^n$. For $\alpha = \langle a_1, \dots, a_n \rangle \in A$, and $\beta = \langle b_1, \dots, b_n \rangle \in B$ define,

$$\alpha \circledast \beta = \langle a_1 \wedge b_1, \dots, a_n \wedge b_n \rangle$$

i.e. the bit-wise conjunction of α and β .

The linear mapping $R: B_n \rightarrow M_{A,B}$ is given by,

$$R(f)_{\alpha, \beta} = f(\alpha \circledast \beta) \quad \alpha \in A, \beta \in B$$

Lemma 5.11:

i) If $f \leq \neg T_{d+1}^n$ and $\#_1(\alpha \oplus \beta) > d$, for each $\alpha \in A$, $\beta \in B$, then $R(f) = \mathbf{0}$, the zero matrix.

ii) $\text{rank}(R) \leq 1$.

Proof: (i) is obvious. For (ii) let $\zeta \in A$ and $\xi \in B$. $R_{\zeta, \xi} = 1$ if and only if,

$$\delta_\alpha \leq \delta_{\zeta \oplus \xi}$$

Therefore the row of $R(\gamma_\alpha)$ indexed by ζ consists entirely of 0's unless $\delta_\alpha \leq \delta_\zeta$. If $\delta_\alpha \leq \delta_\zeta$ then the column of R indexed by ξ contains the value $\gamma_\alpha(\xi)$ regardless of the value of ζ . It follows that all non-zero rows of R are identical establishing (ii). \square

Lemma 5.12: Let $A = B = \{\alpha \in \{0, 1\}^n : \#_0(\alpha) = s\}$, for some parameter s to be fixed. If $f \in B_n$ which satisfies, $\forall \alpha \in \{0, 1\}^n$

$$\begin{aligned} \#_0(\alpha) = s &\Leftrightarrow f(\alpha) = 1 \\ s < \#_0(\alpha) \leq 2s &\Leftrightarrow f(\alpha) = 0 \end{aligned} \tag{5.40}$$

then $R(f)$ has full row rank $\binom{n}{s}$.

Proof: For each $\alpha \in A$, $\beta \in B$, $R_{\alpha, \beta} = 1$ if and only if $\alpha = \beta$. \square

The significance of the linear mapping, R , is due to the fact that there is a close connection between row rank and distance. Paterson (1986) describes an elegant method of establishing this by exploiting the properties of a simple linear transformation, $T: B_n \rightarrow B_n$. Using the notation $\alpha \leq \beta$ as a shorthand for $a_i \leq b_i \forall 1 \leq i \leq n$, where

$$\alpha = \langle a_1, \dots, a_n \rangle, \beta = \langle b_1, \dots, b_n \rangle \in \{0, 1\}^n$$

T is defined by

$$T(\alpha) = \bigoplus_{\beta \leq \alpha} f(\beta) \tag{5.41}$$

The useful properties of this transformation are summarised in,

Lemma 5.13:

$$i) T = \gamma_\alpha$$

$$ii) T = \delta_\alpha$$

$$iii) \forall f \in \mathcal{B}_n, T = f$$

$$iv) f \in P(d) \Leftrightarrow T \leq \neg T_{d+1}^n$$

Proof:

i) $T(\zeta) = \bigoplus_{\beta \leq \zeta} \delta_\alpha(\beta)$. Thus $T(\zeta) = 1$ if and only if $\alpha \leq \zeta$. The only prime implicant of this function is the monom $\bigwedge_{i: a_i=1} x_i = \gamma_\alpha$.

$$\begin{aligned} ii) T(\zeta) &= \bigoplus_{\xi \leq \zeta} \gamma_\alpha(\xi) \\ &= \bigoplus_{\alpha \leq \xi \leq \zeta} 1 = \delta_\alpha(\zeta) \end{aligned}$$

The last equality holds since the interval $\{\xi: \alpha \leq \xi \leq \zeta\}$ is either empty or contains exactly 2^t members. $t=0$ in the latter case if and only if $\zeta = \alpha$.

iii) Immediate from (i) and (ii) using the linearity of T .

iv) From (i) and (ii) T transforms the product $\delta_\alpha(\mathbf{X}_n)$ to the monom, obtained by deleting negated literals, $\gamma_\alpha(\mathbf{X}_n)$; in the reverse direction T replaces the monom γ_α by the product δ_α . Thus if $g \in P(d)$ then $g = \bigoplus_{i=1}^r m_i$, where m_i is a monom containing at most d variables, hence

$$T = \bigoplus_{i=1}^r T$$

where $T(m_i) = \delta_{\alpha_i}$ for some α_i . Each of these products is 0 for any assignment containing more than d 1's, and so $T \leq \neg T_{d+1}^n$. \square

Lemma 5.14: T_{n-s}^n is such that T satisfies (5.40) of Lemma(5.12).

Proof: $T(\alpha) = \bigoplus_{\beta \leq \alpha} T_{n-s}^n(\alpha)$. If $\#_0(\alpha) = s$ then every $\beta \leq \alpha$ has $\#_0(\beta) \geq s$ with equality if and only if $\alpha = \beta$. Thus each $\beta < \alpha$ has $\#_1(\beta) < n - s$ and so $T_{n-s}^n(\beta) = 0$ unless $\beta = \alpha$. If $\#_0(\alpha) > s$, then $\#_1(\alpha) < n - s$ and thus, $T_{n-s}^n(\beta) = 0$ for every $\beta \leq \alpha$ in this case. \square

Theorem 5.11: Let $l = \lfloor k/2 \rfloor$. If $2s + r^l < n$ then

$$BD_k^*(T_{n-s}^n) \geq \binom{n}{s} 2^{r-n}$$

Proof: Let T_{n-s}^n be written as,

$$T_{n-s}^n = \bigoplus_{\phi \in \Phi} \delta_\phi \oplus g$$

where $g \in P(r^l)$.

Note that any $f \in B_n$ can always be written in this form; simply use the identity $f \equiv (\bigoplus_{\alpha: f(\alpha)=1} \delta_\alpha) \oplus 0$.

With this, it follows that

$$\min_{h \in P(r^l)} |T_{n-s}^n \oplus h| \geq \min |\Phi| \tag{5.42}$$

Now consider T , we have from Lemma(5.13) (i) and (iv),

$$T = \bigoplus_{\phi \in \Phi} \gamma_\phi \oplus g'$$

where $g' \leq \neg T_{r^l+1}^n$.

Thus, since $R = R \oplus R$, from Lemma(5.11)(i),

$$R = \bigoplus_{\phi \in \Phi} R$$

and now from Lemma(5.11)(ii), Lemma(5.12) and Lemma(5.14) it

follows that $|\Phi| \geq \binom{n}{s}$. (5.42) and Corollary(5.1) now yield the theorem. \square

Corollary 5.2: For all constant, $k \geq 3$,

$$BD_k^*(MAJ_n) = \exp(\Omega(n^{1/(k-2)}))$$

Proof: Let r, s, l be as in Theorem(5.11) and set $n = 2m - 2s$. Consider a minimal Λ_k formula, S , realising MAJ_n . Under the partial assignment, π , which sets $x_i = 0$ for each $1 \leq i \leq m - 2s$, $S|_{\pi}$ computes the function T_{m-s}^m . Thus,

$$BD_k^*(MAJ_n) \geq BD_k^*(T_{m-s}^m)$$

Note that $2l - 1 \geq k - 2$. In Theorem(5.11) fix $r = \lfloor m^{1/(2l-1)} \rfloor$ and $s = \lfloor (m - r^l - 1)/2 \rfloor$. From the theorem, $BD_k^*(T_{m-s}^m) \geq \binom{m}{s} 2^{r-m}$.

For $p = \frac{1}{2} - \frac{s}{m}$, it holds,

$$\log \left\{ \binom{m}{s} 2^{r-m} \right\} = r - 2mp^2 / \log_e 2 + O(mp^3)$$

and this is $\Omega(r)$.

It follows that,

$$BD_k^*(T_{m-s}^m) \geq \exp(\Omega(m^{1/(k-2)}))$$

Now the Corollary is established by observing that $m \geq \frac{n}{1 + \varepsilon}$ for some constant $\varepsilon > 0$, thus $BD_k^*(MAJ_n) \geq \exp(\Omega(n^{1/(k-2)}))$ also. \square

5.6) Bibliographic Notes

Although it has not been discussed extensively above there is a considerable literature covering the realisation of Boolean functions by DNF. The use of Programmable Logic Arrays (PLAs) as a method of building complex VLSI systems has led to some revival of interest in this area. The classical DNF minimisation algorithms are those described by Karnaugh (1953), McCluskey (1956) and Quine (1952, 1955). Other approaches are presented in Andreev (1983, 1984), Gimpel (1965), Kuznetsov (1983b), Nguen (1982) and (Rhyne et al., 1977). Zhuravlev and Kogan (1985) consider DNF for functions with large numbers of implicants. Arevalo and Bredeson (1978) and Young and Muroga (1979) discuss issues relevant to PLA design. Zhuravlev (1979) considers certain algorithms utilising DNF representations. Various results on the number of distinct DNFs within certain classes are proved by Chukhrov (1982, 1984). Techniques for estimating the complexity of DNF are presented in Mamatov (1979b) and Sapozhenko (1968).

The ringsum expansion has not attracted the same volume of work, however minimisation techniques for this normal form are considered by (Bioul et al., 1973), (Even et al., 1967), (Fleisher et al., 1983), Jagadeesan and Chuang (1970), Mukhopadhyay and Schmitz (1970), Papakonstantinou (1979), Schmookler (1969) and Saluja and Ong (1979).

The known results on monotone bounded-depth networks have been superseded by the work of Razborov and Hastad; Boppana (1986) and Yao (1983) had proved exponential lower bounds for threshold functions; Valiant (1983) proves similar results for certain clique functions when realised by depth-3 networks.

Smolensky (1986) generalises Razborov (1986) by showing that depth- k networks with \neg , \vee and *mod* p -gates (for p prime) must have exponential size to compute the *mod* r functions for any $r \neq p^m$. Razborov (1987) considers the realisation of Boolean functions, by Λ_k formulae, for functions which are "complex" in the sense that related combinatorial structures associated with the functions have interesting extremal properties. The paper proves the existence of polynomial size formulae for the functions examined.

A more powerful bounded-depth model, in which unbounded fan-in threshold functions provide the basis operations, has been introduced by Parberry and Schnitger (1985). As yet no non-trivial lower bounds have been obtained for this.

A different model, in which arbitrary unbounded fan-in gates are permitted, is considered by Hromkovic (1985) and Chandra et al., (1983). The latter paper proves small superlinear bounds for n -input, n -output prefix functions.

Chapter 6

Planar Networks

En toute chose il faut considérer la fin.

Jean de la Fontaine

Fables, III, 5

Le Renard et le boue

6.1) Introduction

In this, concluding, chapter we examine a network model which imposes an, at first sight, rather artificial restriction: that the networks, consisting of 2-input gates, do not contain any pair of wires which cross. In graph-theoretic terms the undirected graph formed by the nodes and their interconnecting wires is planar.^{a)} In Chapter(4) it was observed that much early work relevant to the study of formula complexity was developed in terms of relay-contact schemes; a now obsolete technology. In contrast to this, the planar network restriction is of interest because of close links between it and the complexity issues pertaining to computational models of a recently proposed technology: VLSI circuits. It is not the aim of this text to consider extensively existing work on VLSI complexity. The reader interested in a detailed examination of this topic should consult Ullman (1984). Background on VLSI circuits may be found in Mead and Conway (1980). We will

a) A more rigorous technical formulation of the term planar network is given subsequently at the start of Section(6.2)

be content to outline the relationship between planar network complexity and one class of VLSI models.

Below, in Section(6.2) some results connecting planar and combinational networks are presented. Section(6.3) considers asymptotically matching upper and lower bounds on the size of planar networks realising any Boolean function. The lower bound is from McColl (1985a); the upper bound from McColl and Paterson (1987). A brief overview of VLSI models and their relation to planar networks is given in Section(6.4) which also examines some lower bound results.

6.2) Relations between Planar and Combinational Complexity

In what follows to avoid unnecessary verbiage we say that v is a *terminal* node of a network S if v is an input or an output of S .

Definition 6.1: Let S be an Ω -network ($\Omega \subseteq B_2$) computing some function $f \in B_n$ with $I = \langle i_1, \dots, i_n \rangle$ the set of input nodes of S and t the unique output node. Suppose π is a finite region of the plane with π bounded by a simple, closed curve γ . An *embedding*, ρ , of S onto π is specified by 2 mappings, *PLACE* and *ROUTE*. *PLACE* is an injective mapping associating each node, v , of S with some point $PLACE(v)$ on π ; if v is a terminal node then $PLACE(v)$ must lie on the bounding curve γ , otherwise $PLACE(v)$ must be properly contained within π , i.e not on the boundary. *ROUTE* is also injective and maps wires $\langle v, w \rangle$ of S onto simple connected curves in π in such a way that $ROUTE(\langle v, w \rangle)$ has one endpoint located at $PLACE(v)$ and one endpoint located at $PLACE(w)$. With these two exceptions, $ROUTE(\langle v, w \rangle)$ contains no point α such that $PLACE(u) = \alpha$ for any node u of S .

If $\zeta = \langle z_1, z_2, \dots, z_{n+1} \rangle$ is a given ordering of the terminal nodes of S , then an embedding ρ *respects* ζ if the terminal nodes of

S occur in the cyclic order given by ζ for $\rho(S)$. •

Definition 6.2: Let S be an Ω -network and ζ be the ordering $\langle x_1, \dots, x_n, t \rangle$ of the terminal nodes of S . S is a *planar Ω -network* if and only if there exist a region π with boundary γ , as in Definition(6.1), and an embedding $\rho=(PLACE, ROUTE)$ of S onto π which respects ζ and such that: for all distinct pairs of wires $\langle v, w \rangle$ and $\langle h, u \rangle$ in S the curves $PLACE(\langle v, w \rangle)$ and $PLACE(\langle h, u \rangle)$ have no points of π in common, except possibly endpoints. •

With this formalism established we can introduce the particular complexity measures which are considered in this chapter.

$\mathbf{PC}_\Omega(S)$ denotes the number of gates in the planar Ω -network S . For $f \in B_n$

$$\mathbf{PC}_\Omega(f) = \min\{\mathbf{PC}_\Omega(S) : S \text{ realises } f\}$$

If f is not computable by a planar Ω -network then the quantity $\mathbf{PC}_\Omega(f)$ is undefined. As previously if $\Omega=B_2$ then we use simply $\mathbf{PC}(S)$ and $\mathbf{PC}(f)$ to denote these measures. Similarly in this case we refer to planar networks rather than planar Ω -networks.

With the exception of the following proof, any planar network will be considered as already being embedded to conform to the definition above. For any gate, g , of a planar network we distinguish the two nodes supplying the inputs of g as the *left* input node, $Left(g)$, and the *right* input node, $Right(g)$. $Left(g)$ is found by considering g as a single output gate and rotating the output wire clockwise; $Left(g)$ is the first input of g encountered.

The following result establishes two important facts: that $\mathbf{PC}(f)$ is always well defined, i.e every $f \in B_n$ can be computed by a planar network; and that $\mathbf{PC}(f)$ is "not much greater" than $\mathbf{C}(f)$.

Theorem 6.1: (Lipton and Tarjan, 1980) For all $f \in B_n$, $\mathbf{PC}(f) = O(\mathbf{C}(f)^2)$.

Proof:^{b)} Let $f \in B_n$ and S be a minimal combinational network realising f . Consider the following, not necessarily planar, embedding of S onto the real plane.

Partition the nodes of S into levels $L_0 = \langle x_1, \dots, x_n \rangle, \dots, L_{\mathbf{D}}$ as described in Chapter(1).

The *PLACE* component of the embedding maps nodes, v of S , to points (x_v, y_v) of \mathbf{R}^2 . All nodes v in level L_i are mapped to positions (x_v, y_i) . In this $y_i > y_{i+1}$ for all $0 \leq i < \mathbf{D}(S)$ and for $i=0$ the input nodes have the x co-ordinate set so as to respect the left to right ordering $\langle x_1, \dots, x_n \rangle$. The wires of S are embedded as straight lines connecting nodes. It is assumed that the x -coordinates of nodes are configured so that no embedded wire intersects with an embedded node (other than at endpoints) and that for any given point of \mathbf{R}^2 which is not the image of a node, at most 2 wires cross it.

It should be clear that the embedding described above can be constructed for any network S . This embedding has the following properties.

- E1) A simple closed curve, γ , can be drawn so that all terminal nodes lie on γ , these occurring in the prescribed cyclic order. Furthermore S is embedded onto the region π enclosed by γ .
- E2) Since all wires are straight line segments, any pair of wires have at most one point in common, other than end-points.

(E1) and our construction establish that an embedding respecting the correct cyclic order has been defined. This embedding may not be

b) Lipton and Tarjan (1980) does not give as pedantic a description of the embedding process as our proof does. The reason for the detailed presentation is explained following the theorem proof.

planar, however since S contains exactly $2C(f)$ wires it follows from (E2) that there are at most

$$C(f)(2C(f)-1) = O(C(f)^2)$$

pairs of wires which cross. If we re-impose direction on the embedded wires then a typical crossing appears similar to that depicted in Figure(6.1).

Crossing Environment

Figure 6.1

To complete the proof it suffices to show how any crossing may be simulated by a small planar network. This is accomplished by the network of Figure(6.2).

From the properties of \oplus it is clear that $res(g_v) = res(u)$ and $res(g_u) = res(v)$ in Figure(6.2).

Now noting that all edges in the embedding are directed "downwards", since it has been constructed to respect the partition of S into levels, the action of replacing each crossing by the crossover network

*Planar Crossover Network***Figure 6.2**

of Figure(6.2) does not introduce any directed cycles. The new embedding, which results by replacing each crossing pair, is a planar realisation of some combinational network, T , computing f . Thus,

$$\begin{aligned}\mathbf{PC}(f) &\leq \mathbf{PC}(T) \leq 3\mathbf{C}(f)(2\mathbf{C}(f)-1) + \mathbf{C}(f) \\ &= 6\mathbf{C}(f)^2 - 2\mathbf{C}(f) \\ &= O(\mathbf{C}(f)^2)\end{aligned}$$

□

To see why some care must be exercised in constructing the initial embedding of S used in the theorem proved above, consider the sub-network embedded as in Figure(6.3).

Figure 6.3

Suppose the crossing at α is directly simulated by the network of Figure(6.2) without any other changes being made to the embedding. This would result in the scheme depicted in Figure(6.4).

The new sub-network, although clearly planar, is not admissible since the nodes $\langle a, g_d, d \rangle$ constitute a directed cycle. Thus in constructing a planar realisation it is not sufficient merely to remove all crossings from an arbitrary embedding because this may introduce cycles.

One could define an alternative model, superficially similar to the planar restriction, in which the cost of a given network embedding

Figure 6.4

includes the total number of crossing pairs as well as the number of gates. Formally given a network S and $\rho = (PLACE, ROUTE)$, an embedding of S a complexity measure \mathbf{X}_Ω is defined as:

$$\mathbf{X}_\Omega(S, \rho) = \mathbf{C}_\Omega(S) + |\{(\underline{v}, \underline{w}) : ROUTE(\underline{v}) \text{ crosses } ROUTE(\underline{w})\}|$$

In this \underline{v} and \underline{w} are distinct wires in S . For $f \in B_n$

$$\mathbf{X}_\Omega(f) = \min_{\rho} \min_S \{ \mathbf{X}_\Omega(S, \rho) : S \text{ computes } f \}$$

Obviously $\mathbf{X}_\Omega(f) \leq \mathbf{PC}_\Omega(f)$. Noting behaviour such as that depicted in Figures(6.3, 6.4) McColl (pers. comm.) has posed the following question.

Open Problem: Does there exist any family, $[f_n]$, of Boolean functions for which $\mathbf{PC}_\Omega(f)$ is defined and such that

$$\mathbf{X}_\Omega(f) = o(\mathbf{PC}_\Omega(f)) ? \quad \bullet$$

No examples of this are known, but it has yet to be proved that $\mathbf{X}_\Omega(f) = \Theta(\mathbf{PC}_\Omega(f))$, even in the case $\Omega = B_2$.

Theorem(6.1) uses a basis of 2-input \oplus -gates to implement a planar crossover network; a planar network with ordered inputs $\langle x, y \rangle$ and ordered outputs $\langle y, x \rangle$. McColl (1981) completely characterises those bases $\Omega \subseteq B_2$ from which such crossover networks can be constructed.

Theorem 6.2: (McColl, 1981) A planar crossover can be constructed from a basis $\Omega \subseteq B_2$ if and only if at least one of the following holds:

- i) Ω is complete.
- ii) $\Omega \cap \{\oplus, \iff\} \neq \emptyset$.
- iii) $\{\wedge, \Rightarrow, \Leftarrow\} \subseteq \Omega$.
- iv) $\{\vee, \bar{\Rightarrow}, \bar{\Leftarrow}\} \subseteq \Omega$.

Proof: (Outline) It is easy to verify that any basis satisfying one of the conditions above permits a planar realisation of \oplus or \iff . This can then be used in the scheme of Figure(6.2) to construct a planar crossover. The proof of necessity reduces to considering the sets

$$\begin{aligned} S_1 &= \{\wedge, \vee, \bar{\Rightarrow}\} \\ S_2 &= \{\wedge, \vee, \bar{\Leftarrow}\} \\ S_3 &= \{\wedge, \bar{\Rightarrow}, \bar{\Leftarrow}\} \\ S_4 &= \{\wedge, \vee, \Rightarrow\} \\ S_5 &= \{\wedge, \vee, \Leftarrow\} \\ S_6 &= \{\vee, \Rightarrow, \Leftarrow\} \end{aligned}$$

where for each k ($1 \leq k \leq 6$) and any $\Omega \subseteq S_k$ it may be shown that a planar crossover is not constructible. The argument used to establish this considers the ordered sequences of 2-input functions that may be obtained as outputs of planar networks over such bases and proves that given the ordered inputs $\langle x, y \rangle$ the order cannot be reversed in any admissible output sequence. \square

6.3) Bounds on Planar Network Complexity

As with the network forms examined in earlier chapters asymptotically matching upper and lower bounds on

$$\mathbf{PC}(B_n) = \max \{ \mathbf{PC}(f) : f \in B_n \}$$

have been proved. This section presents both of these results, which in combination establish that

$$\mathbf{PC}(B_n) = \Theta(2^n) \tag{6.1}$$

the lower bound holding for almost all n -input functions. Comparing this lower bound with the upper bounds of Theorem(2.7) and Theorem(4.2) it may be seen that in general planar networks are less efficient than either combinational networks or formulae. This result is from McColl (1985a) and employs Shannon's counting argument in conjunction with a technique for concisely encoding planar networks containing exactly m gates. In deriving this we can restrict attention to \hat{B}_n ; those $f \in B_n$ which are non-degenerate.

In order to improve the lower bound on planar network size implied by Theorem(2.6) we need to show that there are significantly fewer planar networks containing exactly m gates than there are combinational networks of size m . To accomplish this McColl (1985a) defines the following relations between nodes in a planar network.

Let S be a planar network, with unique output node t , and let N be the set of nodes in S . For v, w in N we say that $v \twoheadrightarrow w$ if and only if $v = w$ or there is a directed path from v to w in S . Now suppose that v, w are in N but neither $v \twoheadrightarrow w$ nor $w \twoheadrightarrow v$ hold. In this case define,

$$M(v, w) = \{ m \in N : v \twoheadrightarrow m \text{ and } w \twoheadrightarrow m \}$$

For any appropriate v and w , $M(v, w) \neq \emptyset$ since $t \in M(v, w)$. Furthermore, since S is planar it cannot contain any subgraph homeomorphic to that of Figure(6.5).

Figure 6.5

So there is a unique gate $\mu(v, w) \in M(v, w)$ for which

$$v \twoheadrightarrow \mu(v, w) ; w \twoheadrightarrow \mu(v, w) ; \forall m \in M(v, w) \quad \mu(v, w) \twoheadrightarrow m$$

We can now define an ordering relation \mathfrak{E} over the nodes, N of S . For v, w in N $v \mathfrak{E} w$ if and only if (6.2) or (6.3) below hold.

$$v \twoheadrightarrow w \tag{6.2}$$

$$\left[\begin{array}{c} \neg(v \twoheadrightarrow w \text{ or } w \twoheadrightarrow v) \\ \text{and} \\ v \twoheadrightarrow \text{Left}(\mu(v, w)) \\ \text{and} \\ w \twoheadrightarrow \text{Right}(\mu(v, w)) \end{array} \right] \quad (6.3)$$

Lemma 6.1: The relation \mathbb{E} totally orders the nodes, N , of any planar network S .

Proof: Exercise. \square

It is immediate from this lemma that we can assign to each node, u , of an m -gate planar network S , a unique number, $\lambda(u)$, with $1 \leq \lambda(u) \leq n + m$, where

$$\lambda(u) = |\{v \in N : v \mathbb{E} u\}|$$

Clearly $\lambda(t) = n + m$.

Definition 6.3: Let N be the set of nodes in an n -input, m -gate planar network S . The *RL-specification* of S is the sequence of $n + m - 1$ ordered pairs,

$$\langle R_1, L_1 \rangle ; \cdots ; \langle R_{n+m-1}, L_{n+m-1} \rangle$$

where

$$\begin{aligned} R_i &= |\{v \in N : \lambda(\text{Right}(v)) = i\}| \\ L_i &= |\{v \in N : \lambda(\text{Left}(v)) = i\}| \end{aligned} \quad \bullet$$

Informally R_i (L_i) is the total number of gates for which the node labelled i by λ supplies the right (left) input. Obviously for each $1 \leq i \leq n + m - 1$ we have $R_i + L_i > 0$ since every node, except t , has fan-out at least 1. In addition since every gate has exactly one left input and exactly one right input it holds that

$$\sum_{i=1}^{n+m-1} R_i = \sum_{i=1}^{n+m-1} L_i = m \quad (6.4)$$

Lemma 6.2: Any *RL*-specification describes the graph structure of at most one planar network.

Proof: Suppose the contrary. Let P be the *RL*-specification of 2 different $n + m$ node graph structures, G and H say. Define $G(k)$ to be the structure consisting of the nodes:

$$\lambda^{-1}(1), \dots, \lambda^{-1}(k)$$

from G and the left-right ordered sequence of directed edges leaving these nodes. Should an edge leaving a node in this set enter a gate not in $G(k)$ then the other endpoint is labelled r (l) according to whether the edge forms the *Right* (*Left*) input of its destination gate. The structure $H(k)$ is defined analogously. We adopt the convention of defining $G(0)$ and $H(0)$ as the empty graph.

Since G and H have distinct graph structures there must be some value k , $1 \leq k \leq n + m$, for which

$$G(k-1) = H(k-1) \quad ; \quad G(k) \neq H(k)$$

To prove the lemma it is sufficient to show that $G(k)$ is solely determined by $G(k-1)$ and the *RL*-specification P . For then, since $G(0) = H(0)$, we have $G(k) = H(k)$ for each $0 \leq k \leq n + m$, contradicting the assumption that G and H are distinct. Note that $G = G(n + m)$ and $H = H(n + m)$.

To obtain $G(k)$ from $G(k-1)$ the node $\lambda^{-1}(k)$ has to be added. Suppose that the ordered sequence of edges directed out of $G(k-1)$ contains a consecutive pair labelled $\langle l, r \rangle$. From the definition of \mathfrak{E} the leftmost such pair from the input edges for the node $\lambda^{-1}(k)$, which is a gate. Therefore $G(k)$ is formed by adding this

gate and an ordered sequence of $R_k + L_k$ edges directed out of it. These edges do not enter any gate in $G(k)$ and are labelled, in order, with the sequence

$$\langle r, r, \dots, r, l, l, \dots, l \rangle$$

there being R_k r 's and L_k l 's. If no appropriate pair of edges is present then, again from the definition of \mathfrak{E} , $\lambda^{-1}(k)$ is an input node which is placed to the right of $G(k-1)$, together with $R_k + L_k$ outgoing edges labelled as before, to form $G(k)$.

The argument above applies equally to $H(k)$. Hence we conclude that starting from $G(0) = H(0)$ and following the procedure above, using P , results in two identical graph structures

$$G = G(n + m) = H(n + m) = H \quad \square$$

Theorem 6.3: (McColl, 1985a) For almost all $f \in B_n$, for all $\varepsilon > 0$ and n sufficiently large,

$$\mathbf{PC}(f) \geq \left(\frac{1}{8} - \varepsilon\right) 2^n$$

Proof: We proceed by counting the number of distinct planar networks with n inputs and exactly m gates. A planar network being completely specified by describing its graph structure and the operation associated with each gates, it follows that $RL(n, m) 16^m$, where $RL(n, m)$ is the number of different RL -specifications for n -input m -gate planar networks, is an upper bound on this quantity.

Any RL -specification may be viewed as a pair of partitions of m into $n + m - 1$ non-negative integers, cf (6.6) above. The total number of such partitions is $\binom{n + 2m - 2}{m}$ hence

$$RL(n, m) \leq \binom{n+2m-2}{m}^2$$

It follows that the number of distinct planar networks with n inputs and at most M gates does not exceed

$$\sum_{m=0}^M \binom{n+2m-2}{m}^2 16^m$$

Since $M \gg n$ this quantity is asymptotically

$$\binom{2M}{M}^2 16^M \leq \frac{c 2^{8M}}{\sqrt{M}}$$

So if $M \leq \left(\frac{1}{8} - \varepsilon\right) 2^n$ then the total number of different networks is $o(|B_n|)$ and this proves the theorem. \square

By using a more detailed counting argument McColl (1985a) shows that the constant factor $1/8$ may be increased to $\log_\beta 2$ where

$$\beta = \frac{5(\sqrt{5}-1)}{18-8\sqrt{5}}. \quad \log_\beta 2 \text{ being } 0.172618\dots$$

Savage (1981) describes an upper bound of $(11/2)2^n$ on $\mathbf{PC}(f)$, asymptotically matching the lower bound of Theorem(6.3). We now present a slightly better, although not optimal, construction based on ideas mentioned in McColl (1985a).

Given any $f \in B_n$ with arguments $\mathbf{X}_n = \langle x_1, \dots, x_n \rangle$ define functions g_0 and g_1 in B_{n-1} with arguments $\langle x_1, \dots, x_{n-1} \rangle$ by,

$$\begin{aligned} g_0 &= f^{x_n:=0}(x_1, \dots, x_{n-1}) \\ g_1 &= g_0 \oplus f^{x_n:=1}(x_1, \dots, x_{n-1}) \end{aligned}$$

It is trivial to verify that for any $f \in B_n$

$$f = g_0 \oplus (g_1 \wedge x_n) \quad (6.5)$$

We may employ relation (6.5) as the basis of a recursive construction yielding a planar network for f . This is that network which results by simulating each crossing in the network of Figure(6.6) with the crossover previously described.

Figure 6.6

Theorem 6.4: For all $f \in B_n$, $\mathbf{PC}(f) \leq 4 \cdot 2^n$.

Proof: Consider the network of Figure(6.6). Assuming that the sub-functions, $f^{x_n:=0}$ and $f^{x_n:=1}$ are both realised by planar networks, using the construction recursively, the only crossings that occur are those which arise by placing the inputs for the networks computing these subfunctions. The total number of crossings arising in this way is exactly $\sum_{k=1}^{n-2} k = (n-1)(n-2)/2$. Each of these crossings is simulated by the 3 gate crossover network.

Let a_n denote the total number of gates in the final network; obviously $\mathbf{PC}(f) \leq a_n$. From Figure(6.6) and the fact that each crossing is simulated by a 3 gate network, it is immediate that the recurrence relation below describes the behaviour of a_n .

$$a_n = 2 a_{n-1} + \frac{3(n-1)(n-2)}{2} + 3$$

or equivalently

$$a_n - 2a_{n-1} = \frac{3(n-1)(n-2)}{2} + 3 \quad (6.6)$$

Relations such as (6.6) have a general solution of the form,

$$a_n = A 2^n + c_2 n^2 + c_1 n + c_0 \quad (6.7)$$

For some constants A , c_2 , c_1 and c_0 .

Combining (6.6) with (6.7) and equating coefficients of n^k , for $0 \leq k \leq 2$ we obtain

$$c_2 = c_1 = -3/2 \quad ; \quad c_0 = -6$$

Hence,

$$a_n = A 2^n - \frac{3n^2}{2} - \frac{3n}{2} - 6 \quad (6.8)$$

To obtain the value of A note that $a_2 = 1$; any 2-input Boolean function can be realised using a single gate, the resulting network clearly being planar. Substituting this in (6.8) gives $A = 4$ hence,

$$a_n = 4 \cdot 2^n - \frac{3n^2}{2} - \frac{3n}{2} - 6$$

and hence $\mathbf{PC}(f) \leq 4 \cdot 2^n$ as claimed. \square

By computer analysis McColl and Paterson (1987) establish that for all $f \in B_4$, $\mathbf{PC}(f) \leq 10$. By combining this new boundary condition with a more sophisticated recursive implementation of relation (6.5) they obtain the best upper bound known to date: $\mathbf{PC}(f) \leq (61/48)2^n$.

In summary the results of McColl (1985a) and (McColl and Paterson, 1987) establish that

$$0.172 \cdot 2^n \leq \mathbf{PC}(B_n) \leq 1.271 \cdot 2^n$$

6.4) Planar Networks and VLSI Circuits

Thompson (1980) pioneered the systematic study of VLSI complexity, presenting a formal model of VLSI circuits, defining complexity measures based upon this and developing techniques for proving lower bounds on these with respect to computationally interesting functions such as sorting. The work of Brent and Kung (1981), Vuillemin (1980) and (Lipton and Sedgewick, 1981) produced variants of Thompson's model and gave rise to further lower bound results.

The largest of these bounds were quadratic, in the number of inputs and outputs, and referred to the product AT^2 ; here A is some measure of the area (amount of silicon) required to realise the function by a VLSI chip; T an interpretation of the time taken. However the torrent of $AT^2 = \Omega(n^2)$ results produced between 1980 and 1981

has not continued to the present and now this particular stream of VLSI theory appears to have dried up.

One can attempt to account for the current lack of interest in such results in a number of ways; that existing methods are incapable of producing lower bounds of $\omega(n^2)$ and no progress has been made in constructing more powerful approaches affords only a partial explanation - we have seen that a similar block exists in proving superlinear lower bounds on combinational complexity but there has continued to be some activity within the complexity theory of Boolean networks - however two other reasons could be proposed. Firstly, research into VLSI complexity theory has diversified over a number of related areas e.g the study of formal models of parallel computation (see Gibbons and Rytter (1988) for a survey of this field), although arguably this is a consequence rather than a cause. The second reason may be the close connection exhibited between AT^2 lower bounds and lower bounds on planar network size in Savage (1981). Savage, building on work of Lipton and Tarjan (1979, 1980), unified a number of results on VLSI complexity and established that any lower bound on planar network size held also for the measure AT^2 . In many cases the network lower bound was easier to derive; compare the bound on integer multiplication given below with the sophisticated analysis required to obtain the AT^2 bound directly (Brent and Kung, 1981).^{c)}

A VLSI chip may be viewed as a set of ν layers, each layer containing gates and wires. If two wires cross they must lie on different layers. A constant parameter called the *minimum feature width*, denoted λ , controls certain characteristics of the embedding of gates and wires into layers: each wire must be of width at least λ and

c) This comparison is slightly unfair since Brent and Kung produce a lower bound on the product $AT^{2\alpha}$ for all $\alpha \geq 0$; Savage correlates Area-Time products with planar network size only for $\alpha = 1$ in this case.

separated from any other wire by a space of width λ . Any gate occupies area at least λ^2 . The chip realises a sequential machine, thus the output of a gate at some time t may provide the input of some other gate at time $t+1$. For the computation of a function $f \in B_{n,m}$ we assume a chip has exactly n input ports and m output ports; the input values being supplied exactly once. A denotes the chip area, thus $A = p\lambda^2$ for some value p . T denote the number of iterations taken for the computation to complete. Savage (1981) describes how an equivalent combinational network can be constructed by breaking feedback loops and using T copies of a chip in sequence; the inputs for the $t+1$ copy being supplied by the outputs of gates on the t 'th copy. From this method we have,

Theorem 6.5: (Savage, 1981) For all $f \in B_{n,m}$,

$$\mathbf{C}(f) \leq v(A/\lambda^2)T(f) \quad \square$$

Theorem 6.6: (Savage, 1981) For all $f \in B_{n,m}$

$$\mathbf{PC}(f) = O(AT^2(f))$$

Proof: (Outline) Given any VLSI circuit realising f in area A and T iterations the T copies, in the construction described above, may be embedded so that the total number of crossing wires is $O(AT^2)$. In combination with Theorem(6.5) this gives the upper bound. \square

Theorem(6.6) shows that lower bounds on the Area-Time product can be deduced from lower bounds on planar network size. To conclude this chapter we describe a general approach to proving such bounds and illustrate its application to integer multiplication.

An important tool in all existing lower bound approaches is the following well-known result due to Lipton and Tarjan (1979).

Theorem 6.7: (Planar Separator Theorem) Let $G(V, E)$ be an n -vertex planar graph in which each vertex $v \in V$ is assigned a non-negative cost, $c(v)$, and let $wt(G) = \sum_{v \in V} c(v)$. There is a partition of V into 3 sets, A , B and C satisfying the following properties.

$$\text{i) } \sum_{v \in A} c(v) \leq \frac{2 \, wt(G)}{3}.$$

$$\text{ii) } \sum_{v \in B} c(v) \leq \frac{2 \, wt(G)}{3}.$$

$$\text{iii) } |C| \leq \sqrt{8n}.$$

iv) A and B are *separated* by C ; that is every path from a vertex in A to a vertex in B must go through some vertex in C .

Lipton and Tarjan (1980) described how this result could be applied to obtain quadratic bounds on the planar network complexity of certain m -output functions.

Suppose we wish to prove a lower bound for some $f \in B_{n,m}$. Consider any planar network realising f . Assign unit cost to some subset V of the input nodes and some subset W of the output nodes. To all other nodes assign cost 0. Applying the planar separator theorem we can subsequently identify collections $X' \subseteq V$, $Y' \subseteq W$ which are separated by some set of gates C . Now $|C| \leq \sqrt{8(\mathbf{PC}(f) + n)}$ and if some subfunction of f , $g: X' \rightarrow Y'$ has sufficiently many points in the image of its domain then we may be able to apply the pigeon-hole principle to argue that $|C| = \Omega(n)$. If this is so then it follows that $\mathbf{PC}(f) = \Omega(n^2)$.

Of course in order for this approach to be successful the structure of f must be such that an appropriate subfunction, g , can always be identified regardless of which variables result in X' and Y' .

A class of suitable functions has been identified by Savage.

Definition 6.4: Let $f \in B_{n,m}$ with input variables \mathbf{X}_n and outputs $\mathbf{Y}_m = \langle y_1, \dots, y_m \rangle$. Let $V \subseteq \mathbf{X}_n$ and $W \subseteq \mathbf{Y}_m$. f has a w -flow (with respect to $V \cup W$) if and only if: for all partitions of $V \cup W$ into 2 sets A and B such that

$$|A| \leq \frac{2(|V|+|W|)}{3} \quad ; \quad |B| \leq \frac{2(|V|+|W|)}{3}$$

there exist $X' \subseteq A \cap V$ and $Y' \subseteq B \cap W$ (or $X' \subseteq B \cap V$ and $Y' \subseteq A \cap W$) such that for some assignment, π to $\mathbf{X}_n - X'$ the subfunction $g = f|_{\pi} : X' \rightarrow Y'$ has at least 2^w points in the image of its domain.

$f \in B_{n,m}$ has a w -flow if appropriate subsets V and W exist. •

Theorem 6.8: (Savage, 1981) If $f \in B_{n,m}$ has a w -flow then

$$\mathbf{PC}(f) \geq \frac{w^2}{8} - O(n+m)$$

Proof: Let $f \in B_{n,m}$ have a w -flow with respect to subsets V of \mathbf{X}_n and W of \mathbf{Y}_m . Consider an optimal planar network, T , realising f . It is convenient to make a minor modification to T in order to simplify the proof; arrange that every input node x_i of V has fanout exactly one, every output node y_j of W has fanin exactly one and that every path from any such input to any such output contains at least one other node. These alterations can be carried out using at most $n+m$ extra gates. S will denote the network resulting by modifying T in this way.

Assign unit cost to each node in $V \cup W$ and a cost of zero to every other node in S . Applying the Planar Separator Theorem partitions the nodes of S into 3 sets A'' , B'' and C'' which satisfy

$$|A'' \cap (V \cup W)| \leq \frac{2|V \cup W|}{3} \quad (6.9)$$

$$|B'' \cap (V \cup W)| \leq \frac{2|V \cup W|}{3} \quad (6.10)$$

$$|C''| \leq \sqrt{8(\mathbf{PC}(S) + n)} \quad (6.11)$$

In addition A'' and B'' are separated by C'' . $A'' \cup B''$ may not contain every node in $V \cup W$ since some of these may have been allocated to C'' . Let $U = C'' \cap (V \cup W)$. From the way S has been constructed each $u \in U$ has either fanin 1, if it is an output, or fanout 1, if it is an input node. Rearrange the partition so that the nodes in U are distributed over A'' and B'' to preserve (6.9) and (6.10). This rearrangement may result in A'' and B'' no longer being separated, however this can only happen if the unique input (output), v , of some node $u \in U$ is in A'' and u is assigned to B'' or vice-versa. The separation property can now be restored by moving v to C'' . Let A' , B' and C denote the new partition which results after moving U out of C'' and additional nodes, as necessary, into C'' . Clearly (6.9), (6.10) and (6.11) still hold for this new separating partition since $|C| \leq |C''|$. Additionally let

$$A = A' \cap (V \cup W) \quad ; \quad B = B' \cap (V \cup W)$$

A and B form a partition of $V \cup W$ satisfying the conditions of Definition(6.4).^{d)}

d) The restructuring of T is required to ensure that the partition just described can be constructed. Lipton and Tarjan (1980) proves lower bounds directly without this modification, however the analysis used in the final stages of their proofs becomes more complicated as a result.

S realises f and f has a w -flow, therefore, without loss of generality, we can find $X' \subseteq A \cap V$, $Y' \subseteq B \cap W$ and a partial assignment π fixing $\mathbf{X}_n - X'$ so that the subfunction $g = f^{\pi} : X' \rightarrow Y'$ has at least 2^w points in the image of its domain. Consider the sub-network of S^{π} containing only those nodes which are ancestors of nodes in Y' . In this sub-network, since X' and Y' are separated, every path from a node in X' must encounter a gate in C . It follows that if $\langle z_1, z_2, \dots, z_{|C|} \rangle$ is some ordering of the gates in C then over all assignments to X' there are at most $2^{|C|}$ distinct values that this sequence can take. However the values taken by these gates completely determine the values at the outputs Y' . It follows that $2^{|C|} \geq 2^w$ thus $|C| \geq w$. Hence from (6.11) and our earlier observation that $|C| \leq |C''|$ it follows that $\mathbf{PC}(f) \geq w^2/8 - O(n+m)$ as required. \square

Definition 6.5: Let $f \in B_{n+s,n}$ which has n data inputs, $\langle x_1, \dots, x_n \rangle$; $s \geq 0$ shifting inputs, $\langle z_1, \dots, z_s \rangle$; and n outputs $\langle y_1, \dots, y_n \rangle$. f is a *shifting function* if for all $0 \leq k \leq n$ there is an assignment to the shifting inputs with which $y_{i+k} := x_i$ for each $1 \leq i \leq n-k$. •

Theorem 6.9: Let $f \in B_{n+s,n}$ be a shifting function. f has an $(n/18)$ -flow.

Proof: (Below we assume n is even). Let $V = \{x_1, \dots, x_{n/2}\}$ and $W = \{y_{n/2+1}, \dots, y_n\}$. Consider any partition of $V \cup W$ into two sets, A and B , satisfying

$$|A| \leq \frac{2n}{3} \quad ; \quad |B| \leq \frac{2n}{3}$$

Let $A_X = A \cap V$, $A_Y = A \cap W$, $B_X = B \cap V$ and $B_Y = B \cap W$. Without loss of generality it may be assumed that $|A| \leq |B|$ and $|A_X| \geq |A_Y|$. With these assumptions we therefore have

$$\frac{n}{3} \leq |A| \leq \frac{n}{2}$$

and so $|A_X| \geq \frac{n}{6}$. In addition,

$$|B_Y| = n - |A| - |B_X| \geq \frac{n}{6}$$

Given $x_p \in A_X$ and $y_q \in B_Y$ there is exactly one value r in the range $1 \leq r \leq n/2$ for which $q = p + r$. We call the triple $\langle p, q, r \rangle$ in this case a *match*. Since $|A_X| \geq n/6$, $|B_Y| \geq n/6$ the total number a possible matches is at least $n^2/36$. However there are only $n/2$ choices for the value of r so there must be some value l , $1 \leq l \leq n/2$ which forms a match with at least $n/18$ pairs of indices in $A_X \times B_Y$. Thus for this value of l we can find $k \geq n/18$ inputs $X' = \langle x_{i_1}, \dots, x_{i_k} \rangle \subseteq A_X$ for which

$$Y' = \langle y_{i_1+l}, \dots, y_{i_k+l} \rangle \subseteq B_Y$$

Since f is a shifting function and none of the shifting inputs occur in V the variables $\langle z_1, \dots, z_s \rangle$ may be fixed to realise a shift by l places, the remaining inputs, apart from X' , can be fixed arbitrarily. The resulting subfunction $g: X' \rightarrow Y'$ has exactly $2^{|X'|}$ points in the image of its domain and since $|X'| \geq n/18$ this proves the theorem. \square

Corollary 5.1: If $f \in B_{n+s,n}$ is a shifting function then

$$\mathbf{PC}(f) \geq \frac{n^2}{648} - O(n) \quad \square$$

Corollary 5.2: Let $MULT \in B_{2n,2n-1}$ be the integer multiplication function.

$$\mathbf{PC}(MULT) \geq \frac{n^2}{648} - O(n)$$

Proof: The first n inputs of *MULT* may be regarded as data inputs, the remaining n inputs as shifting inputs. The sub-function formed by considering only the n least significant bits of the $(2n - 1)$ -bit output tuple defines a shifting function. \square

Bibliographic Notes

The proof of Theorem(6.3) can be readily adapted to show $\mathbf{X}(f) = \Omega(2^n)$ for almost all $f \in B_n$. The "proof" in Mamatov (1975) which purports to establish $\mathbf{X}_{\{\wedge, \vee, \neg\}}(f) = O(2^n/n)$ contains a number of serious errors. McColl (1985a) further shows that Theorem(6.3) continues to hold for more general forms of planar network in which multiple copies of inputs are permitted, provided the total number of input nodes allowed is $o(2^n/\log n)$. When $O(2^n/\log n)$ inputs nodes are used the formula size upper bound of Theorem(4.2) applies.

Savage (1981) also improves the lower bound on the planar complexity of Boolean Matrix Product originally given by Lipton and Tarjan (1980), as well as transforming VLSI lower bounds of Vuillemin (1980) to planar networks. Also in the field of VLSI complexity, Kramer and Van Leeuwen (1983) obtain an analogue of Lupanov's results on combinational complexity for VLSI circuits.

Planar monotone networks were first considered in McColl (1985b). There an exact value for the planar monotone complexity of T_2^n is obtained. It is also proved that T_k^n for $3 \leq k \leq n - 2$ cannot be computed by networks (of 2-input gates) which are simultaneously monotone and planar. Beynon and Buckle (1987) describe an effective procedure which decides if any given $f \in M_n$ is planar monotone computable. Their results establish that

$$\mathbf{PC}_{\{\wedge, \vee\}}(f) = O(n^4)$$

for any $f \in M_n$ for which this is defined. The best lower bound for this measure is

$$\mathbf{PC}_{\{ \wedge, \vee \}}(T_2^n) = \Omega(n^2)$$

from McColl (1985b). At present no Shannon style counting argument has been discovered.

Ἐν δ' ἔπεσ' Ὠχρανὼ λαμπρὸν φῶς ἠελίοιο,
Ἐλχὸν νυχτὰ μελαιναν ἔπι ζειδωρὸν ἄρουραν

Iliad, *viii*, 485-6

Bibliography

- [1] Abullaev, D.A; Yunosov, D: (1975) Symmetrical Boolean function decomposition; Avtom. i Telemekh, 2, 12-13 (In Russian)
- [2] Aho, A.V; Hopcroft, J.E; Ullman, J.D: (1974) The design and analysis of computer algorithms; Addison-Wesley
- [3] Ajtai, M: (1983) Σ_1^1 -formulae on finite structures; Ann. Pure and Appl. Logic, 24, 1-48
- [4] Ajtai, M; Ben-Or, M: (1984) A theorem on probabilistic constant depth computations; Proc. 16th ACM Symposium on Theory of Computing, 471-474
- [5] Ajtai, M; Komlos, J; Szemerédi, E: (1983) An $O(n \log n)$ sorting network; Proc. 15th ACM Symposium on Theory of Computing, 1-9
- [6] Alekseev, V.B: (1973) On the number of k -valued monotonic functions; Doklady Akademii Nauk SSSR, 208, 505-508 (In Russian) (Transl: Sov. Math.-Doklady, 14, 87-91)
- [7] Alekseev, V.B: (1976) The decipherment of certain classes of monotone multivalued functions; Z. Vysicl. Mat. Fiz., 16, 189-198 (In Russian)
- [8] Alon, N; Boppana, R: (1986) The monotone circuit complexity of Boolean functions; Combinatorica, 7, 1-22
- [9] Alt, H: (1984) Comparison of arithmetic functions with respect to Boolean circuit depth; Proc. 16th ACM Symposium on Theory of Computing, 466-470
- [10] Andreev, A.E: (1983) On the synthesis of disjunctive normal forms which are close to minimal; Sov. Math.-Doklady, 27, 265-269
- [11] Andreev, A.E: (1984) On the problem of minimising disjunctive normal forms; Sov. Math.-Doklady, 29, 32-36
- [12] Andreev, A.E: (1985) A method of proving lower bounds on the complexity of monotone Boolean functions; Doklady Akademii Nauk SSSR, 282, 1033-1037 (In Russian) (Transl: Sov. Math.-Doklady, 31, 530-534)
- [13] Andreev, A.E: (1986) A method of proving superquadratic lower bounds on the complexity of Π -schemes; Vestn. Moscow Un-ta Series 1, Matematika Mechanika, No. 6, 73-75 (In Russian)
- [14] Andreev, A.E: (1987) A method of proving effective lower bounds on monotone complexity; Algebra and Logic, T.26, 1, 3-26 (In Russian)
- [15] Arevalo, Z; Bredeson, J.G: (1978) A method to simplify a Boolean function into a near minimal sum-of-products for programmable logic arrays; IEEE Trans. Comput., C-27, 1028-1030
- [16] Aslanjan, L.A: (1975) A method of recognition that is based on division into classes by disjunctive normal forms; Kibernetika, 5, 103-110 (In Russian) (Transl: Cybernetics, 11, (1976), 779-787)
- [17] Avgustinovich, S.V: (1980) An approach to obtaining lower bounds on complexity for Boolean functions; Metody Diskret. Analiz., 35, 104 (In Russian)
- [18] Avizienis, A: (1961) Signed-digit number representations for fast parallel arithmetic; IRE Trans. Electron. Computers, EC-10, 389-400
- [19] Babai, L; Pudlak, P; Rodl, V; Szemerédi, E: (1987) Lower bounds to the complexity of symmetric Boolean functions; Internal Report

- [20] Baker, T; Gill, J; Solovay, R: (1975) Relativizations of the $P = ? NP$ question; SIAM Jnl. on Computing, 4, 161-173
- [21] Baker, T; Selman, A: (1975) A second step toward the Polynomial Hierarchy; Proc. 17th IEEE Symposium on Foundations of Computer Science
- [22] Barak, A; Shamir, E: (1976) On the parallel evaluation of Boolean expressions; SIAM Jnl. on Computing, 5, 678-681
- [23] Batchner, K.E: (1968) Sorting networks and their applications; Proc. AFIPS Spring Joint Computer Conf., 32, 307-314
- [24] Beame, P.W; Cook, S; Hoover, H.J: (1984) Log-depth circuits for division and related problems; Proc. 25th IEEE Symposium on Foundations of Computer Science, 1-6
- [25] Berge, C: (1979) Graphs and Hypergraphs; North-Holland
- [26] Berkowitz, S: (1982) On some relationships between monotone and non-monotone circuit complexity; Technical Report, Univ. of Toronto
- [27] Beynon, W.M: (1984) Replaceability and computational equivalence for monotone Boolean functions; Acta Informatica, 22, 433-449
- [28] Beynon, W.M; Buckle, J: (1987) Monotone Boolean functions computable by planar circuits; Theoretical Computer Science, 53, 267-279
- [29] Bini, D; Pan, V. Ya: (1987) A logarithmic Boolean time algorithm for parallel polynomial division; Inf. Proc. Letters, 24, 233-237
- [30] Bioul, G; Davio, M; Deschamps, J.P: (1973) Minimization of ring-sum expansions of Boolean functions; Philips Res. Rep., 28, 17-36
- [31] Bloniarz, P: (1979) The complexity of monotone Boolean functions and an algorithm for finding shortest paths in a graph; Ph.D Dissertation; Technical Report No. 238, Lab. for Computer Science, MIT
- [32] Blum, N: (1984a) A Boolean function requiring $3n$ network size; Theoretical Computer Science; 28, 337-345
- [33] Blum, N: (1984b) An $\Omega(n^{4/3})$ lower bound on the monotone network complexity of n -th degree convolution; Theoretical Computer Science, 36, 59-70
- [34] Blum, N; Seysen, M: (1984) Characterization of all optimal networks for a simultaneous computation of AND and NOR; Acta Informatica, 21, 171-181
- [35] Boppana, R; Lagarias, J.C: (1986) One-way functions and circuit complexity; Structure In Complexity Theory, Springer-Verlag, Lecture Notes in Computer Science, 223, 51-65
- [36] Boppana, R: (1986) Threshold functions and bounded-depth monotone circuits; Jnl. of Comp. and Syst. Sci., 32, 222-229
- [37] Born, R.C; Scidmore, A.K: (1968) Transformation of switching functions to completely symmetric switching functions; IEEE Trans. Computers, C-17, 596-599
- [38] Borodin, A: (1977) On relating time and space to size and depth; SIAM Jnl. on Computing, 6, 733-744
- [39] Borodin, A; Munro, I: (1975) The computational complexity of algebraic and numeric problems; American Elsevier, New York
- [40] Breitbart, Y.Y: (1968) Comparison of the complexities of realisation of Boolean functions by Automata and Turing machines; Doklady Akademii Nauk SSSR, 180,

- 1053-1055 (In Russian) (Transl: Sov. Phys.-Doklady, 13, 524-526)
- [41] Breitbart, Y.Y; Reiter, B: (1975a) Algorithms for fast evaluation of Boolean expressions; *Acta Informatica*, 4, 107-116
 - [42] Breitbart, Y.Y; Reiter, B: (1975b) A branch-and-bound algorithm to obtain an optimal evaluation tree for monotonic Boolean functions; *Acta Informatica*, 4, 311-319
 - [43] Breitbart, Y.Y; Gal, S: (1978) Analysis of algorithms for the evaluation of monotonic Boolean functions; *IEEE Trans. Comput.*, C-27, 1083-1087
 - [44] Bremer, H: (1974) Upper and lower bounds for the complexity of Boolean functions; Springer-Verlag, *Lecture Notes in Computer Science*, 33, 99-102
 - [45] Brent, R; Kuck, D.J; Maruyama, K: (1973) The parallel evaluation of arithmetic expressions without division; *IEEE Trans. Computers*, C-22, 532-534
 - [46] Brent, R; Kung, H.T: (1981) The Area-Time complexity of binary multiplication; *Jnl. of the ACM*, 28, 521-534
 - [47] Brown, W.G: (1966) On graphs that do not contain a Thompson graph; *Canadian Math. Bull.*, 9, 281-285
 - [48] Brustmann, B; Wegener, I: (1986) The complexity of symmetric functions in bounded-depth circuits; Preprint
 - [49] Bublitz, S: (1986) Decomposition of graphs and monotone formula size of homogeneous functions; *Acta Informatica*, 23, 689-696
 - [50] Chandra, A.K; Fortune, S; Lipton, R: (1983) Lower bounds for constant depth circuits for prefix problems; *Proc. 10th ICALP*, Springer-Verlag, *Lecture Notes In Computer Science*, 154, 109-117
 - [51] Chandra, A.K; Stockmeyer, L.J; Vishkin, U: (1984) Constant depth reducibility; *SIAM Jnl. on Computing*, 13, 423-439
 - [52] Chiang, K-W; Vranesic, Z.G: (1983) A tree representation of combinational networks; *IEEE Trans. Comput.*, C-32, 3, 315-319
 - [53] Chukhrov, I.P: (1982) On the number of irredundant disjunctive normal forms; *Sov. Math. Dokl.*, 25, 254-257
 - [54] Chukhrov, I.P: (1984) On the number of minimal disjunctive normal forms; *Sov. Math. Dokl.*, 29, 714-718
 - [55] Church, R: (1940) Numerical analysis of certain free distributive structures; *Duke Math. Jnl.*, 6, 732-734
 - [56] Commentz-Walter, B: (1979) Size-depth tradeoff in monotone Boolean formulae; *Acta Informatica*, 12, 227-243
 - [57] Commentz-Walter, B; Sattler, J: (1980) Size-depth tradeoff in non-monotone Boolean formulae; *Acta Informatica*, 14, 257-269
 - [58] Cook, S.A: (1971) The complexity of theorem proving procedures; *Proc. 3rd ACM Symposium on Theory of Computing*, 151-158
 - [59] Cook, S.A: (1974) An observation on time-storage trade-off; *Jnl. of Computer and System Sciences*, 9, 308-316
 - [60] Cook, S.A: (1979) Deterministic CFLs are accepted simultaneously in polynomial time and log squared space; *Proc. 11th ACM Symposium on Theory of Computing*, 338-345

- [61] Cook, S.A: (1981) Towards a complexity theory of synchronous parallel computation; *L'Enseignement Mathématique*, 1-2, 99-124
- [62] Cook, S.A; Hoover, H.J: (1985) A depth-universal circuit; *SIAM Jnl. on Computing*, 14, 833-839
- [63] Dedekind, R: (1897) *Über Zerlegungen von Zahlen durch ihre grössten gemeinsamen Teiler*; Reprinted in: *Ges. Math. Werke II*, Chelsea, N.Y (1969), 103-108
- [64] Denenberg, L; Gurevich, Y; Shelah, S: (1983) Cardinalities definable by constant-depth, polynomial-size circuits; Report TR-26-83, Aiken Computation Lab., Harvard University
- [65] Dudich, V.N: (1973) Synthesis of schemes of switching contacts; *Kibernetika*, 9, 21-25 (In Russian) (Transl: *Cybernetics*, 9, 392-396)
- [66] Dunne, P.E: (1984a) Techniques for the analysis of monotone Boolean networks; Ph.D Dissertation; Theory of Computation Report No.69, Univ. Of Warwick,
- [67] Dunne, P.E: (1984b) Lower bounds on the monotone complexity of threshold functions; In: Proc. of 22nd Annual Allerton Conf. on Communication, Control and Computing, 911-920
- [68] Dunne, P.E: (1984c) Some Results On Replacement Rules In Monotone Boolean Networks; Theory Of Computation Report No.64, Univ. Of Warwick
- [69] Dunne, P.E: (1985a) A $2.5n$ lower bound on the monotone network complexity of T_3^n ; *Acta Informatica*, 22, 229-240
- [70] Dunne, P.E: (1985b) On monotone simulations of non-monotone networks; Report CSR 85/7 Dept. of Computer Science, Univ. of Liverpool
- [71] Dunne, P.E: (1985c) Approximate replacement rules and pseudo-complementation; Report CSR 85/9 Dept. of Computer Science, Univ. of Liverpool
- [72] Dunne, P.E: (1986) The complexity of central slice functions; *Theoretical Computer Science*, 44, 247-257
- [73] Dunne, P.E: (1987) Sympathetic bases and the complexity of realising Boolean functions by networks involving non-monotone operations; Report CSR 87/4, Dept. of Computer Science, Univ. of Liverpool
- [74] Edenbrandt, A: (1987) Chordal graph recognition is in NC ; *Inf. Proc. Letters*, 24, 239-241
- [75] Ehrenfeucht, A: (1975) Practical decidability; *Jnl. of Computer and System Sciences*, 11, 392-396
- [76] Elspas, B; Kautz, H.W; Stone, H.W: (1968) Properties of modular multifunctional computer networks; Stanford Res. Inst, Menlo Park, California, Project 4641 (AFRCL), Final report
- [77] Erdős, P; Spencer, J: (1974) *Probabilistic methods in combinatorics*; Academic Press, New York
- [78] Even, S: (1979) *Graph Algorithms*; Pitman
- [79] Even, S; Kohavi, I; Paz, A: (1967) On minimal modulo 2 sums of products for switching functions; *IEEE Trans. Electron. Computers*, 16, 671-674
- [80] Fagin, R; Klawe, M; Pippenger, N.J; Stockmeyer, L: (1985) Bounded-depth, polynomial size circuits for symmetric functions; *Theoretical Computer Science*, 36, 239-250

- [81] Finikov, B.I: (1957) On a family of classes of functions in the logic algebra and their realisation in the class of Π -schemes; Doklady Akademii Nauk SSSR, 115, 247-248 (In Russian)
- [82] Fischer, M: (1974) The complexity of negation-limited networks; Springer-Verlag, Lecture Notes in Computer Science 33, 71-82
- [83] Fischer, M; Meyer, A; Paterson, M.S: (1982) $\Omega(n \log n)$ lower bounds on the length of Boolean formulas; SIAM Jnl. on Computing, 11, 416-426
- [84] Fischer, M; Pippenger, N.J: (1979) Relations among complexity measures; Jnl. of the ACM, 26, 361-381
- [85] Fischer, M; Rabin, M.O: (1974) Super-exponential complexity of Presburger arithmetic; In: Complexity of Computation (Ed: R.M.Karp)
- [86] Fleisher, H; Tavel, M; Yeager, J.D: (1983) Exclusive-Or representation of Boolean functions; IBM Jnl. Res. Dev., 27, 412-416
- [87] Friedman, A.D: (1975) Logical Design Of Digital Systems; Pitman
- [88] Friedman, J: (1986) Constructing $O(n \log n)$ size monotone formulae for the k -th threshold function of n Boolean variables; SIAM Jnl. on Computing, 15, 641-654
- [89] Furst, M; Saxe, J.B; Sipser, M: (1984) Parity, circuits and the polynomial Time hierarchy; Math. Syst. Theory; 17, 13-27
- [90] Galbiati, G; Fischer, M: (1981) On the complexity of 2-output Boolean networks; Theoretical Computer Science, 16, 177-185
- [91] Galil, Z; Paul, W: (1983) An efficient general purpose parallel computer; Jnl. of the ACM, 2, 360-387
- [92] Garey, M; Johnson, D: (1979) Computers and intractability - a guide to the theory of NP-completeness; Freeman
- [93] Gaskov, S.B: (1978) The depth of Boolean functions; Problemy Kibern., 34, 265-268 (In Russian)
- [94] Gaskov, S.B: (1980) The complexity of realisation of Boolean functions by schemes and formulas in bases consisting of continuous functions; Sov. Math. Doklady, 21, 186-190
- [95] Gavrillov, M.A; Kuznetsov, O.P; Khazotskii, V.E: (1969) Description and analysis of switching circuits with large numbers of input variables; Avtom. i Telemekh, 16, 108-115 (In Russian) (Transl: Autom. Remote Control, 16, 1643-1650)
- [96] Gershkovich, Y.B; Polterovich, V.M: (1967) Nonrepeating superpositions of Boolean functions of two variables; ibid, 5, 753-760 (In Russian) (Transl: ibid, 5, 109-152)
- [97] Gibbons, A.M; Rytter, W: (1988) Efficient parallel algorithms; Cambridge University Press
- [98] Gilbert, E.N: (1954) Lattice theoretic properties of frontal switching functions; Jnl. Math. and Phys., 33, 57-97
- [99] Gimpel, J.F: (1965) A method of producing a Boolean function having an arbitrarily described prime implicant table; IEEE Trans. Computers, 14, 484-488
- [100] Greene, C; Kleitman, D: (1976) Strong versions of Sperner's theorem; Jnl. Combinatorial Theory, Series A, 20, 80-88
- [101] Grigoriev, D.Y: (1976) Using the concepts of separability and independence to obtain lower bounds on the complexity of circuits; Zap. Nauk. Sem. Leningrad

- Otdel Mat. Inst. Steklov (LOMI), 60, 221-222 (In Russian)
- [102] Gurevich, I.B; Zhuravlev, Y.U: (1974) Minimisation of Boolean functions and effective recognition algorithms; *Kibernetika*, 10, 16-20 (In Russian) (Transl: *Cybernetics*, 10, 393-7)
- [103] Hansel, G: (1964) Nombre minimal de contacts de fermeture necessaires pour realiser une fonction Booleene symmetrique de n variables; *C.R Acad. Sci., Parise, Groupe 1*, 6037-6040
- [104] Hansel, G: (1966a) Sur le nombre des fonctions Booleenes monotones de n variables; *ibid*, 262 Series A, 1088-1090
- [105] Hansel, G: (1966b) Construction d'un schema de contacts bipolaire pour une fonction Booleene isotone arbitraire de n variables; *ibid*, 263 Series A, 651-654
- [106] Harper, L.H: (1975) A note on some classes of Boolean functions; *Stud. Appl. Math*, 54, 161-164
- [107] Harper, L.H; Hsieh, W.N; Savage, J.E: (1975) A class of Boolean functions with linear combinational complexity; *Theoretical Computer Science*, 1, 161-183
- [108] Harper, L.H; Savage, J.E: (1972) On the complexity of the marriage problem; *Advances in Mathematics*, 9, 299-312
- [109] Harrison, M.A: (1965) *Introduction to switching and automata theory*; McGraw-Hill
- [110] Hastad, J: (1986) Almost optimal lower bounds lower bounds for small depth circuits; *Proc. 18th ACM Symposium on Theory of Computing*, 6-20
- [111] Hennie, F.C; Stearns, R.E: (1966) Two-tape simulation of multitape Turing machines; *Jnl. of the ACM*, 13, 533-546
- [112] Hodes, L: (1970) The logical complexity of geometric properties in the plane; *Jnl. of the ACM*, 17, 339-347
- [113] Hodes, L; Specker, E: (1968) Lengths of formulas and elimination of quantifiers I; in *Contributions to Mathematical Logic*, H.A.Schmidt, K.Schutte and H.-J.Thiele (editors); North-Holland, 175-188
- [114] Hoover, H.J; Klawe, M.M; Pippenger, N.J: (1984) Bounding fan-out in logical networks; *Jnl. of the ACM*, 31, 13-18
- [115] Hopcroft, J.E; Karp, R.M: (1973) An $n^{5/2}$ algorithm for maximum matching in bipartite graphs; *SIAM Jnl. on Computing*, 2, 225-231
- [116] Hopcroft, J.E; Ullman, J.D: (1979) *Introduction to automata theory, languages and computation*; Addison-Wesley
- [117] Hromkovic, J: (1985) Linear lower bounds on unbounded fan-in Boolean circuits; *Inf. Proc. Letters*, 21, 71-74
- [118] Huynh, D.T: (1987) On solving hard problems by polynomial size circuits; *Inf. Proc. Letters*, 24, 171-176
- [119] Hyafil, L: (1976) Bounds for selection; *SIAM Jnl. on Computing*, 5, 109-114
- [120] Hyafil, L; Kung, H.T: (1975) The complexity of parallel evaluation of linear recurrences; *7th ACM Symposium on Theory of Computing*, 12-22
- [121] Jagadeesan, M; Chuang, Y.H: (1970) Minimization of Boolean functions in mod-2 sum of products form; In: *1970 SW IEECO IEEE Conf., Rec.*, 473-477
- [122] Jerrum, M; Snir, M: (1982) Some exact complexity results for straight-line computations over semirings; *Jnl. of the ACM*, 29, 874-897

- [123] Jukna, S.P: (1986) Lower bounds on the complexity of local circuits; Proc. 12th Symp. on Mathematical Foundations of Comp. Sci., Springer-Verlag, Lecture Notes in Computer Science, 233, 440-448
- [124] Jukna, S.P: (1987) Computations in lattices of partitions; Mathematical Logic and its Applications, No. 5, 17-21
- [125] Jung, H: (1985) Depth efficient transformations of arithmetic into Boolean circuits; FCT '85, Springer-Verlag, Lecture Notes in Computer Science, 199, 167-174
- [126] Karatsuba, A; Ofman, Y: (1962) Multiplication of multidigit numbers on automata; Doklady Akademii Nauk SSSR, 145, 293-294 (In Russian) (Transl: Sov. Phys.-Doklady, 7 (1963), 595-596)
- [127] Karchmer, M; Wigderson, A: (1987) Monotone circuits for connectivity require super-logarithmic depth; Internal Report, Hebrew Univ., Jerusalem
- [128] Karnaugh, M: (1953) The map method for synthesis of combinational logic circuits; Trans. AIEE, 72, 593-598
- [129] Karpova, N.A: (1975) Some remarks on the asymptotic behaviour of Shannon's functions; Problemy Kibernet., 30, 313-318 (In Russian)
- [130] Karpovski, M.G; Moskalev, E.S: (1967) Realisation of a system of logical functions by means of an expansion in orthogonal series; Avtom. i Telemekh, 12, 119-129 (In Russian) (Transl: Autom. Remote Control, 12, 1921-1931)
- [131] Kasim-Zade, O.M: (1980a) A measure of complexity of networks composed of functional elements; Doklady Akademii Nauk SSSR, 250, 797-800 (In Russian)
- [132] Kasim-Zade, O.M: (1980b) A measure of the complexity of schemes of functional elements; Sov. Math. Dokl., 21, 203-206
- [133] Khasin, L.S: (1969a) Complexity bounds for the realisation of monotonic symmetrical functions by means of formulas in the basis $\{\wedge, \vee, \neg\}$; Doklady Akademii Nauk SSSR, 189, 752-755 (In Russian) (Transl: Sov. Phys. Dokl., 14 (1970), 1149-1151)
- [134] Khasin, L.S: (1969b) On realisations of monotone symmetric functions by formulas in the basis $\{\wedge, \vee, \neg\}$; Problemy Kibernet., 21, 253-257 (In Russian) (Transl: Syst. Theory Res., 21 (1971), 254-259)
- [135] Khrapchenko, V.M: (1963) On a method of transforming a multiseria code into a uniseria one; Doklady Akademii Nauk SSSR, 148, 296-299 (In Russian) (Transl: Sov. Phys.-Doklady, 8, 8-10)
- [136] Khrapchenko, V.M: (1967) Asymptotic estimation of addition time of a parallel adder; Problemy Kibernet., 19, 107-122 (In Russian) (Transl: Syst. Theory Res., 19 (1970), 105-122)
- [137] Khrapchenko, V.M: (1971a) On the complexity of the realisation of the linear function in the class of Π -circuits; Mat. Zametki, 9, 35-40, (In Russian) (Transl: Math. Notes of Academy of Sciences of USSR, 9, 21-23)
- [138] Khrapchenko, V.M: (1971b) Methods of determining lower bounds for the complexity of Π -schemes; *ibid*, 10, 83-92 (In Russian) (Transl: *ibid*, 10, 474-479)
- [139] Khrapchenko, V.M: (1972) The complexity of the realisation of symmetrical functions by formulas; *ibid*, 11, 109-120 (In Russian) (Transl: *ibid*, 11 (1972), 70-76)
- [140] Khrapchenko, V.M: (1976) Complexity of realisation of symmetric algebraic logic functions on finite bases; Problemy Kibernet, 31, 231-234 (In Russian)

- [141] Khrapchenko, V.M: (1978) Depth and delay in a network; Soviet Math., 19, 1006-1009
- [142] Kleitman, M; Pippenger, N.J: (1978) An explicit construction of short monotone formulae for the monotone symmetric functions; Theoretical Computer Science, 7, 325-332
- [143] Kleitman, D: (1969) On Dedekind's problem: the number of monotone Boolean functions; Proc. AMS, 21, 677-682
- [144] Kleitman, D: (1973) The number of Sperner families of subsets of an n element set; Colloq. Math. Soc. Janos Bolyai, 10, Infinite and Finite sets, Keszthely, Hungary, 989-1001
- [145] Kleitman, D; Markowsky, G: (1975) On Dedekind's problem: the number of isotone Boolean functions II; Trans. AMS, 213, 373-390
- [146] Kloss, B.M: (1966) Estimates of the complexity of solutions of systems of linear equations; Doklady Akademii Nauk SSSR, 171, 781-783 (In Russian) (Transl: Sov. Math.-Doklady, 7, 1537-1540)
- [147] Kloss, B.M; Malyshev, V.A: (1965) Estimates of the complexity of certain classes of functions; Vestn. Moskva. Univ. Ser. 1, 4, 44-51 (In Russian)
- [148] Knuth, D.E: (1973) Fundamental algorithms; Addison-Wesley
- [149] Kodanpani, K.L; Seth, S.C: (1978) On combinational networks with restricted fan-out; IEEE Trans. Comput., C-27, 309-318
- [150] Korobkov, V.K: (1965) On monotone functions in the algebra of logic; Problemy Kibernet., 13, 5-28 (In Russian)
- [151] Korshunov, A.D: (1981) On the number of monotone Boolean functions; Problemy Kibernet., 38, 5-108 (In Russian)
- [152] Kramer, M; van Leeuwen, J: (1983) The VLSI complexity of Boolean functions; Springer-Verlag, Lecture Notes in Computer Science, 171, 397-407
- [153] Krichevskii, R.E: (1959) Realisations of functions by superpositions; *ibid*, 2, 123-138 (In Russian) (Transl: Problems of Cybernetics, 2 (1961), 458-477)
- [154] Krichevskii, R.E: (1963) Complexity of contact circuits realising a function of logical algebra; Doklady Akademii Nauk SSSR, 151, 803-806 (In Russian) (Transl: Sov. Phys.-Doklady, 8 (1964), 770-772)
- [155] Kriegel, K; Waack, S: Lower bounds for Boolean formulae of depth 3 and the topology of the n -Cube; FCT '85, Springer-Verlag, Lecture Notes in Computer Science, 199, 227-233
- [156] Kuznetsov, S.E: (1981) Combinatorial circuits with no null chains over the basis $\{\wedge, \vee, \neg\}$; Izvestija VUZ, Matematika, 5, 56-63 (In Russian)
- [157] Kuznetsov, S.E: (1983a) On the complexity of the realisation of a sequence of Boolean functions by formulas of depth 3 in the basis $\{\wedge, \vee, \neg\}$; Ver. Met. Kib.; 19, 40-43 (In Russian)
- [158] Kuznetsov, S.E: (1983b) On the lower estimate of the length of the shortest disjunctive normal form for almost all Boolean functions; *ibid*, 19, 44-47
- [159] Ladner, R.E; Fischer, M: (1980) Parallel prefix computation; Jnl. of the ACM, 27, 831-838
- [160] Lai, H.C; Muroga, S: (1979) Minimal parallel binary adders with NOR (NAND) gates; IEEE Trans. Comput., C-28, 648-659

- [161] Lamagna, E.A: (1979) The complexity of monotone networks for certain bilinear forms, routing problems, sorting and merging; IEEE Trans. Computers, C-28, 773-782
- [162] Lamagna, E.A; Savage, J.E: (1974) Combinational complexity of some monotone functions; Proc. 15th IEEE Symposium on Switching and Automata Theory, 140-144
- [163] Langheld, E: (1976) Introduction to threshold and majority logic; Elektronik, 25, 46-52 (In German)
- [164] Lenz, K; Wegener, I: (1987) The conjunctive complexity of quadratic Boolean functions; Internal Report FB Nr. 240, Abteilung Informatik, Univ. Dortmund
- [165] Levey, S.Y; Paull, M.C: (1969) An algebra with application to sorting algorithms; Proc. Princeton Conf. on Information Sci. and Systems, 285-291
- [166] Lingas, A: (1979) Lower bounds for straight-line algorithms; (from Ph.D Dissertation)
- [167] Lipton, R; Sedgewick, R.E: (1981) Lower bounds for VLSI; Proc. 13th ACM Symposium on Theory of Computing, 300-307
- [168] Lipton, R; Tarjan, R.E: (1979) A separator theorem for planar graphs; SIAM Jnl. on Applied Mathematics, 36, 177-189
- [169] Lipton, R; Tarjan, R.E: (1980) Applications of a planar separator theorem; SIAM Jnl. on Computing, 9, 615-627
- [170] Long, D: (1986) The monotone circuit complexity of threshold functions; Unpublished manuscript; Univ. Of Oxford
- [171] Lupanov, O.B: (1958) On a method of circuit synthesis; Izvestia VUZ (Radiofizika), 1, 120-140 (In Russian)
- [172] Lupanov, O.B: (1959) On the asymptotic bounds of the complexities of formulas which realise logic algebra functions; Doklady Akademii Nauk SSSR, 128, 464-467 (In Russian) (Transl: Autom. Expr., 2 (1960), 12-14)
- [173] Lupanov, O.B: (1960a) The complexity of realising functions of logical algebra by means of formulas; Problemy Kibernet., 3, 61-80 (In Russian) (Transl: Autom. Expr., 3 (1961), 30)
- [174] Lupanov, O.B: (1960b) Complexity of formula realisation of functions of logical algebra; Problemy Kibernet., 3, 61-80 (In Russian) (Transl: Problems of Cybernetics, 3 (1962), 782-811)
- [175] Lupanov, O.B: (1961a) Implementing the algebra of logic functions in terms of bounded depth formulas in the basis Of $\{\wedge, \vee, \neg\}$; Doklady Akademii Nauk SSSR, 136, 1041-1042 (In Russian) (Transl: Sov.-Phys.-Doklady, 6, 107-108)
- [176] Lupanov, O.B: (1961b) On the principle of local coding and the realisation of functions of certain classes of networks composed of functional elements; Doklady Akademii Nauk SSSR, 140, 322-325 (In Russian) (Transl: Sov. Phys.-Doklady, 6, 750-752)
- [177] Lupanov, O.B: (1961c) On the realisation of functions of logical algebra, by formulae of finite classes (formulae of limited depth) in the basis $\{\wedge, \vee, \neg\}$; Problemy Kibernet., 6, 5-14 (In Russian) (Transl: Problems of Cybernetics, 6 (1965), 1-14)
- [178] Lupanov, O.B: (1962a) On comparing the complexity of the realisations of monotonic functions by contact networks containing only closing contacts and by

- arbitrary contact networks; *Doklady Akademii Nauk SSSR*, 144, 1245-1248 (In Russian) (Transl: *Sov. Phys.-Doklady*, 7, 486-489)
- [179] Lupanov, O.B: (1962b) A class of circuits of functional elements; *Problemy Kibernet.*, 7, 61-114 (In Russian)
- [180] Lupanov, O.B: (1965a) The problem of realising symmetric functions in the algebra of logic by contact schemes; *Problemy Kibernet.*, 15, 85-99 (In Russian)
- [181] Lupanov, O.B: (1965b) An approach to systems synthesis - The principle of Local Coding; *Problemy Kibernet.*, 14, 31-110 (In Russian)
- [182] Lupanov, O.B: (1970) Effect of the depth of formulas on their complexity; *Kibernetika*, 2, 46-49 (In Russian) (Transl: *Cybernetics*, 6 (1970), 62-66)
- [183] Lupanov, O.B: (1972) Circuits using threshold elements; *Doklady Akademii Nauk SSSR*, 202, 1282-1291 (In Russian) (Transl: *Sov. Phys.-Doklady*, 17, 91-93)
- [184] Lupanov, O.B: (1973) Complexity of the universal parallel-series network of depth 3; *Trudy Matem. Inst. Steklov*, 133, 127-131 (In Russian)
- [185] Machtey, M; Young, P: (1978) An introduction to the general theory of algorithms; North-Holland
- [186] Madatjan, H.A: (1980) On correction of the totality of recognition algorithms by schemes of functional elements; *Sov. Math. Dokl.*, 22, 687-691
- [187] Malyshev, V.A: (1967) The class of "almost all" functions with nonlinear complexity in the class of Π -networks; *Problemy Kibernet.*, 19, 299-306 (In Russian) (Transl: *Syst. Theory Res.*, 19 (1970), 305-312)
- [188] Mamatov, Y.A: (1975) Asymptotic estimation of the complexity of plane logical grids realising logic functions; *Tekh. Kibernet.*, 13, 135-9 (In Russian) (Transl: *Eng. Cybernetics*, 13, 107-110)
- [189] Mamatov, Y.A: (1979a) Concerning a principle for obtaining lower bounds on the complexity of formulas; *Sov. Math.-Doklady*, 20, 339-342
- [190] Mamatov, Y.A: (1979b) On a principle for obtaining high (exponential for some parameter values) lower bounds for the complexity of disjunctive normal forms; *Sov. Math.-Doklady*, 20, 399-401
- [191] Markov, A.A: (1957) On the inversion complexity of a system of functions; *Doklady Akademii Nauk SSSR*, 116, 917-919 (In Russian) (Transl: *Jnl. of the ACM*, 5 (1958), 331-334)
- [192] Masek, W.J: (1978) Some NP-complete set covering problems; (unpublished manuscript)
- [193] McCluskey, E.J: (1956) Minimization of Boolean functions; *Bell Sys. Tech. Jnl.*, 35, 1417-1444
- [194] McColl, W.F: (1976) The depth of Boolean functions; *Proc. 3rd ICALP*, 307-321
- [195] McColl, W.F: (1977) Some results on circuit depth; Ph.D Dissertation; *Theory of Computation Report No.18*, Dept. of Computer Science, University of Warwick
- [196] McColl, W.F: (1978a) The maximum depth of monotone formulae; *Inf. Proc. Letters*, 7, 65
- [197] McColl, W.F: (1978b) Complexity hierarchies for Boolean functions; *Acta Informatica*, 11, 71-77

- [198] McColl, W.F: (1978c) The circuit depth of symmetric Boolean functions; *Jnl. of Computer and System Sciences*, 17, 108-115
- [199] McColl, W.F: (1981) Planar crossovers; *IEEE Trans. Comput.*, C-30, 223-225
- [200] McColl, W.F: (1985a) Planar circuits have short specifications; *Proc. 2nd Annual Symposium on Theoretical Aspects of Computer Science*, Springer-Verlag, *Lecture Notes in Computer Science*, 182, 231-242
- [201] McColl, W.F: (1985b) On the planar monotone computation of threshold functions; *ibid*, 219-230
- [202] McColl, W.F; Paterson, M.S: (1977) The depth of all Boolean functions; *SIAM Jnl. on Computing*, 6, 373-380
- [203] McColl, W.F; Paterson, M.S: (1987) The planar realization of Boolean functions; *Inf. Proc. Letters*, 24, 165-170
- [204] Mead, C; Conway, L: (1980) *Introduction to VLSI systems*; Addison-Wesley
- [205] Mehlhorn, K: (1976) An improved lower bound on the formula complexity of context-free recognition; *Elektron. Informationsverarbeiten Kybern.*, 12, 523-4 (In German)
- [206] Mehlhorn, K: (1979) Some remarks on Boolean sums; *Acta Informatica*, 12, 371-375
- [207] Mehlhorn, K; Galil, Z: (1976) Monotone switching networks and Boolean matrix product, *Computing*, 16, 99-111
- [208] Meyer, A; Stockmeyer, L: (1973) The equivalence problem for regular expressions with squaring requires exponential space; *Proc 13th IEEE Conf. on Switching and Automata Theory*
- [209] Miller, R.E: (1965) *Switching theory: Volume 1: Combinational circuits*; Wiley
- [210] Mirwald, R; Schnorr, C.P: (1987) The multiplicative complexity of quadratic Boolean forms; *Proc. 28th FOCS*
- [211] Moran, S: (1987) Generalized lower bounds derived from Hastad's Main Lemma; *Inf. Proc. Letters*, 25, 383-388
- [212] Muchnik, B.A: (1970) Bound on complexity of realisation of a linear function by formulas in certain bases; *Kibernetika*, 4, 29-38 (In Russian) (Transl: *Cybernetics*, 4 (1973), 395-406)
- [213] Mukhopadhyay, A; Schmitz, G: (1970) Minimization of Exclusive-Or and Logical Equivalence switching circuits; *IEEE Trans. Comput.*, C-19, 132-140
- [214] Muller, D.E: (1956) Complexity in electronic switching circuits; *IRE Trans. Computers*, EC-5, 15-19
- [215] Muller, D.E; Preparata, F.P: (1975) Bounds to complexities of networks for sorting and for switching; *Jnl. of the ACM*, 22, 195-201
- [216] Muroga, S; Lai, H.C: (1976) Minimization of logic networks under a generalised cost function; *IEEE Trans. Comput.*, C-25, 893-907
- [217] Nakasima, A: (1936) The theory of relay circuits; *Nippon Elec. Communication Engineering*, May, 197-226
- [218] Neciporuk, E.I: (1960) On the complexity of superpositions in bases that contain non-trivial linear formulas with zero weight; *Doklady Akademii Nauk SSSR*, 136, 560-563 (In Russian) (Transl: *Sov. Phys.-Doklady*, 6 (1961), 6-9)

- [219] Neciporuk, E.I: (1961) Complexity of networks in containing non-trivial elements with zero weights; *ibid*, 139, 1302-1303 (In Russian) (Transl: *Sov. Math.-Doklady*, 2, 1087-1088)
- [220] Neciporuk, E.I: (1962a) On the complexity of networks in certain bases containing non-trivial elements with zero weights; *Problemy Kibernet.*, 8, 123-160 (In Russian)
- [221] Neciporuk, E.I: (1963) On the synthesis of logical nets in incomplete and degenerate bases; *Doklady Akademii Nauk SSSR*, 155, 299-301 (In Russian) (Transl: *Sov. Phys.-Doklady*, 9 (1964), 299-301)
- [222] Neciporuk, E.I: (1964a) Synthesis of circuits from threshold elements; *ibid*, 154, 763-766 (In Russian) (Transl: *Sov. Math.-Doklady*, 5, 163-166)
- [223] Neciporuk, E.I: (1964b) On self correcting gating circuits; *ibid*, 156, 1045-1048 (In Russian)
- [224] Neciporuk, E.I: (1964c) The synthesis of networks from threshold elements; *Problemy Kibernet.*, 11, 49-62 (In Russian) (Transl: *Autom. Expr*, 7, 35-39)
- [225] Neciporuk, E.I: (1965) Complexity of gating circuits which are realised by Boolean matrices with undetermined elements; *Doklady Akademii Nauk SSSR*, 163, 40-42 (In Russian) (Transl: *Sov. Phys.-Doklady*, 10 (1966), 591-593)
- [226] Neciporuk, E.I: (1966) A Boolean function; *Doklady Akademii Nauk*, 169, 765-766 (In Russian) (Transl: *Sov. Math.-Doklady*, 7, 999-1000)
- [227] Neciporuk, E.I: (1969) On a Boolean matrix; *Problemy Kibernet.*, 21, 237-240 (In Russian) (Transl: *Syst. Theory Res.*, 21 (1971), 236-239)
- [228] Neciporuk, E.I: (1970) Realisations of disjunctions and conjunctions in monotone bases; *Problemy Kibernet.*, 23, 291-293 (In Russian) (Transl: *Syst. Theory Res.*, 23 (1973), 305-307)
- [229] Nguen, K.A: (1982) On some characteristics of algorithms for minimizing Boolean functions; *Sov. Math.-Doklady*, 26, 741-745
- [230] Nigmatullin, R.G: (1984) The complexity of universal functions and lower bounds on the complexity; *Izvestija VUZ, Matematika*, 11, 10-20 (In Russian)
- [231] Nigmatullin, R.G: (1985) Are lower bounds on the complexity lower bounds for universal circuits?; *FCT '85*, Springer-Verlag, *Lecture Notes in Computer Science*, 199, 331-340
- [232] Ofman, Y: (1962) On the algorithmic complexity of discrete functions; *Doklady Akademii Nauk SSSR*, 145, 48-51 (In Russian) (Transl: *Sov. Phys.-Doklady*, 7 (1963), 589-591)
- [233] Ofman, Y: (1963) Approximate realisation of continuous functions by automata; *ibid*, 152, 823-826 (In Russian) (Transl: *Sov. Math.-Doklady*, 4, 823-826)
- [234] Okol'nishnikova, B: (1982) On the influence of negations on the complexity of a realisation of monotone Boolean functions by formulae of bounded-depth; *Metod. Diskr. Anal.*, 38, 74-80 (In Russian)
- [235] Orlov, V.A: (1971) The algorithmic unsolvability of the problem of finding the asymptotic behaviour of the Shannon function in the realisation of boundedly deterministic operators using networks in an arbitrary basis; *Doklady Akademii Nauk SSSR*, 196, 1036 (In Russian) (Transl: *Sov. Phys.-Doklady*, 16, 81-83)
- [236] Papakonstantinou, G: (1979) Minimization of modulo-2 sum of products; *IEEE Trans. Comput.*, C-28, 163-167

- [237] Parberry, I; Schnitger, G: (1985) Parallel computation with threshold functions; Report CS-85-32, Dept. of Comp. Sci., Penn. State Univ.
- [238] Paterson, M.S: (1975) Complexity of monotone networks for Boolean matrix product; Theoretical Computer Science, 1, 13-20
- [239] Paterson, M.S: (1986) Bounded-depth circuits over $\{\oplus, \wedge\}$; Preprint, Univ. of Warwick
- [240] Paterson, M.S: (1987) Improved sorting networks with $O(\log n)$ depth; Research Report RR89, Univ. of Warwick
- [241] Paterson, M.S; Valiant, L.G: (1976) Circuit size is nonlinear in depth; Theoretical Computer Science, 2, 397-400
- [242] Paterson, M.S; Wegener, I: (1986) Nearly optimal hierarchies for network and formula size; Acta Informatica, 23, 217-221
- [243] Paul, W: (1975) Boolean minimal polynomials and covering problems; Acta Informatica, 4, 321-336
- [244] Paul, W: (1976) Realizing Boolean functions on disjoint sets of variables; Theoretical Computer Science, 2, 383-396
- [245] Paul, W: (1977) A $2.5n$ lower bound on the complexity of Boolean functions; SIAM Jnl. on Computing, 6, 427-443
- [246] Peterson, G.L: (1978) An upper bound on the size of formulae for symmetric Boolean functions; Tech. Report No. 78-03-01, Dept. of Computer Science, Univ. of Washington
- [247] Pippenger, N.J: (1974) Short formulae for symmetric functions; IBM Report RC 5143, Yorktown Heights, NY
- [248] Pippenger, N.J: (1976) The realization of monotone Boolean functions; Proc. 8th ACM Symposium on Theory of Computing, 204-210
- [249] Pippenger, N.J: (1977) Information theory and the complexity of Boolean functions; Math. Sys. Theory, 10, 129-167
- [250] Pippenger, N.J: (1978) The complexity of monotone Boolean functions; Math. Sys. Theory, 11, 289-316
- [251] Pippenger, N.J: (1980) Pebbling with an auxiliary pushdown; IBM Research Report RJ3012
- [252] Pippenger, N.J; Valiant, L.G: (1976) Shifting graphs and their applications; Jnl. of the ACM, 23, 423-432
- [253] Post, E.L: (1941) Two-valued iterative systems of mathematical logic; Annals of Math. Studies, 5, Princeton Univ. Press
- [254] Pratt, V.R: (1975a) The effect of basis on size of Boolean expressions; Proc. 16th IEEE Symposium on FOCS, 119-121
- [255] Pratt, V.R: (1975b) The power of negative thinking in multiplying Boolean matrices; SIAM Jnl. on Computing, 4, 326-330
- [256] Preparata, F.P; Muller, D.E; Barak, A.B: (1977) Reduction of depth of Boolean networks with a fan-in constraint; IEEE Trans. Comput., C-26, 474-479
- [257] Preparata, F.P; Muller, D.E: (1970) Generation of near-optimal universal Boolean functions; Jnl. of Computer and System Sciences, 4, 93-102

- [258] Preparata, F.P; Muller, D.E: (1971) On the delay required to realise Boolean functions; IEEE Trans. Computers, C-20, 459-461
- [259] Preparata, F.P; Muller, D.E: (1976) Efficient parallel evaluation of Boolean expressions; *ibid*, C-25, 548-549
- [260] Pudlak, P: (1983) Bounds for Hodes-Specker Theorem; Logic and Machines: Decision Problems and Complexity (Proceedings); Springer-Verlag, Lecture Notes In Computer Science, 171, 421-445
- [261] Pulatov, A.K: (1979) Lower bounds on the complexity of implementation of characteristic functions of group codes by Π -networks; Combinatorial-Algebraic methods in Applied Mathematics, Gorki, 81-95 (In Russian)
- [262] Pupyrev, E.I: (1977) On finding a redundant subformula in a Boolean function formula; Prob. Control and Inf. Theory, 6, 243-7
- [263] Quine, W.V: (1952) The problem of simplifying truth functions; American Mathl. Monthly, 59, 521-531
- [264] Quine, W.V: (1955) A way to simplify truth functions; American Mathl. Monthly, 62, 627-631
- [265] Ramsey, F.P: (1930) On a problem of formal logic; Proc. London. Mathl. Soc., 30, 264-286
- [266] Razborov, A.A: (1985a) Lower bounds on the monotone complexity of some Boolean functions; Doklady Akademii Nauk SSSR, 281, 798-801; (In Russian) (Transl: Sov. Math. Doklady, 31, 354-357)
- [267] Razborov, A.A: (1985b) A lower bound on the monotone complexity of the logical permanent; Mat. Zametki, 37, 887-901; (In Russian) (Transl: Mathem. Notes of the Acad. of Sci. of the USSR, 37, 485-493)
- [268] Razborov, A.A: (1986) Lower bounds on the complexity of bounded-depth networks over the basis $\{\wedge, \oplus\}$; Preprint, Steklov Institute, Moscow Univ. (In Russian);
Extended abstract in: Uspekhi Mat. Nauk, T.41 No.4 (1986), 219-220 (In Russian); (Transl: Russian Math. Surveys (Comm. Moscow Mathl. Soc. section), 41 No.4 (1986), 181-182)
- [269] Razborov, A.A: (1987) Bounded-depth formulae over the basis $\{\wedge, \oplus\}$ and some combinatorial problems; "Complexity of algorithms and applied mathematical logic", Series: Questions in Cybernetics (In Russian)
- [270] Razborov, A.A: (1988a) On the method of approximations; Preprint, Steklov Institute, Moscow Univ.
- [271] Razborov, A.A: (1988b) An application of matrix methods to the theory of lower bounds on the complexity of computation; Preprint, Steklov Institute, Moscow Univ. (In Russian)
- [272] Red'kin, N.P: (1969a) Synthesis of two-layer threshold-element circuits; Avtom. i Telemekh, 2, 82-91 (In Russian) (Transl: Autom. Remote Control, 2, 233-241)
- [273] Red'kin, N.P: (1969b) Complexity of realisation of incompletely defined Boolean functions; *ibid*, 9, 118-122 (In Russian) (Transl: *ibid*, 3 (1970), 1474-1477)
- [274] Red'kin, N.P: (1970) Decompositional approach to circuit synthesis; *ibid*, 8, 84-88 (In Russian) (Transl: *ibid*, 1273-1277)
- [275] Red'kin, N.P: (1971) Realisation of Boolean functions in a certain class of threshold element circuits; *ibid*, 8, 102-107 (In Russian) (Transl: *ibid* (1972), 1252-1256)

- [276] Red'kin, N.P: (1973) Proof of minimality of circuits consisting of functional elements; Problemy Kibernet., 23, 83-102 (In Russian) (Transl: Syst. Theo. Res, 23, 85-103)
- [277] Red'kin, N.P: (1975) Realization of systems of conjunctions by contact circuits; ibid, 30, 263-76 (In Russian)
- [278] Red'kin, N.P: (1979) On the realisation of monotone Boolean functions by contact circuits; ibid, 35, 87-110 (In Russian)
- [279] Reznik, V.I: (1961) The realization of monotonic functions by means of networks consisting of functional elements; Doklady Akademii Nauk SSSR, 139, 566-569 (In Russian) (Transl: Sov. Phys.-Doklady, 6 (1962), 558-561)
- [280] Reischer, C; Simovici, D: (1984) Graph functions of Boolean functions; IEEE Trans. Comput., C-33, 97-99
- [281] Rhyne, T.V; Noe, P.S; McKinney, M.H; Pooch, U.W: (1977) A new technique for the fast minimization of switching functions; IEEE Trans. Comput., C-26, 757-764
- [282] Riordan, J; Shannon, C.E: (1942) The number of two-terminal series-parallel networks; Jnl. Math. and Phys. 21, 83-93
- [283] Rivest, R.L: (1977) The necessity of feedback in minimal monotone combinational circuits; IEEE Trans. Comput., C-26, 606-607
- [284] Romankevich, H.M; Yatsunov, A.I: (1974) On a method of representing Boolean functions; Avtom. i Telemekh, 3, 30-35 (In Russian)
- [285] Rudich, S; Berman, L: (1987) Optimal circuits and transitive automorphism groups; IBM Research Report RC12688
- [286] Ruzzo, W.L: (1981) On uniform circuit complexity; Jnl. of Computer and System Sciences, 22, 365-383
- [287] Saluja, K.K; Ong, E.H: (1979) Minimization of Reed-Muller canonic expansion; IEEE Trans. Comput., C-28, 535-537
- [288] Sapozhenko, A.A: (1968) On the greatest length of a dead-end disjunctive normal form for almost all Boolean functions; Mat. Zametki, 4, 649-658 (In Russian)
- [289] Sarkisjan, G.Z: (1978) Effective computability of arithmetic predicates and functions on the basis of schemes of functional elements; Izv. Akad. Nauk Armjan. SSSR Ser. Mat., 13, 128-139 (In Russian, English summary)
- [290] Savage, J.E: (1971) The complexity of decoders - Part II Computational work and decoding time, IEEE Trans. Inf. Theory, IT-17, 77-84
- [291] Savage, J.E: (1972) Computational work and time on finite machines, Jnl. of the ACM, 19, 660-674
- [292] Savage, J.E: (1974) An algorithm for the computation of linear forms; SIAM Jnl. on Computing, 3, 150-158
- [293] Savage, J.E: (1976) The complexity of computing; John Wiley
- [294] Savage, J.E: (1981) Planar circuit complexity and the performance of VLSI algorithms; VLSI Systems and Computation. H.T.Kung, B.Sproull and G.Steele (Editors), Computer Science Press, 61-68
- [295] Savitch, W.J: (1970) Relationship between nondeterministic and deterministic tape complexities; Jnl. of Computer and System Sciences, 4, 166-192

- [296] Schmookler, M.S: (1969) On mod-2 sums of products; IEEE Trans. Computers, C-18, 957
- [297] Schnorr, C.P: (1974) Zwei lineare untere schranken fur die komplexitat Boolescher funktionen; Computing, 13, 155-171
- [298] Schnorr, C.P: (1976a) The network complexity and Turing machine complexity of finite functions; Acta Informatica, 7, 95-107
- [299] Schnorr, C.P: (1976b) The combinational complexity of equivalence; Theoretical Computer Science, 1, 289-295
- [300] Schnorr, C.P: (1976c) A lower bound on the number of additions in monotone computations of monotone rational polynomials; Theoretical Computer Science, 2, 305-317
- [301] Schnorr, C.P: (1976d) The network complexity and the breadth of Boolean functions; Logic Colloquium 76, 491-504
- [302] Schnorr, C.P: (1980) A $3n$ -lower bound on the network complexity of Boolean functions; Theoretical Computer Science, 10, 83-92
- [303] Schnorr, C.P: (1986) A Gödel theorem on network complexity lower bounds; Zeitschr. f. math. Logik und Grundlagen d. Math., 32, 377-384
- [304] Schonhage, A; Strassen, V: (1971) Schnelle multiplikation grosser zahlen; Computing, 7, 281-292
- [305] Schurfeld, U: (1983) New lower bounds on the formula size of Boolean functions; Acta Informatica, 19, 183-194
- [306] Sethi, I.K: (1980) Fast sequential evaluation of monotonic Boolean functions; Inf. Sci., 20, 101-113
- [307] Shamir, E; Snir, M: (1980) On the depth complexity of formulas; Math. Syst. Theory, 13, 301-322
- [308] Shannon, C.E (1938) A symbolic analysis of relay and switching circuits; Trans. AIEE, 57, 713-723
- [309] Shannon, C.E: (1949) The Synthesis of two-terminal switching circuits; Bell System Tech. Jnl., 28, 59-98
- [310] Shestakov, V.I: (1938) Some mathematical methods for the construction and simplification of two-terminal electrical networks of class A; Dissertation, Lomonosov State Univ. (Moscow) (In Russian)
- [311] Sholomov, L.A: (1967) On functionals characterising the complexity of a system of undetermined Boolean functions; Problemy Kibernet., 19, 123-140 (In Russian) (Transl: Syst. Theory Res., 19 (1970), 123-141)
- [312] Sholomov, L.A: (1969) On the realization of incompletely defined Boolean functions by circuits of functional elements; *ibid*, 21, 215-226 (In Russian) (Transl: *ibid*, 21 (1972), 211-223)
- [313] Sholomov, L.A: (1970) On calculating the complexity of Boolean functions on Turing machines; *ibid*, 22, 53-66 (In Russian) (Transl: *ibid*, 22 (1972), 51-65)
- [314] Sholomov, L.A: (1971) Information complexity of problems associated with minimal realisation of Boolean functions by networks; Doklady Akademii Nauk SSSR, 200, 556-559 (In Russian) (Transl: Sov. Phys.-Doklady, 16 (1972), 714-717)
- [315] Sipser, M: (1983) Borel sets and circuit complexity; Proc. 15th ACM Symposium on Theory of Computing, 61-69

- [316] Sklansky, J: (1960a) An evaluation of several two-sum and binary adders; IRE Trans. Electron. Computers, EC-9, 213-226
- [317] Sklansky, J: (1960b) Conditional sum addition logic; *ibid*, EC-9, 226-231
- [318] Skyum, S: (1983) A measure in which Boolean negation is exponentially powerful; Inf. Proc. Letters, 17, 125-128
- [319] Skyum, S; Valiant, L.G: (1985) A complexity theory based on Boolean algebra; Jnl. of the ACM, 32, 484-502
- [320] Smolensky, R: (1987) Algebraic methods in the theory of lower bounds for Boolean circuit complexity; Proc. 19th ACM Symposium on Theory of Computing, 77-82
- [321] Soprunenko, E.P: (1965) Minimal realisations of functions by circuits using functional elements; Problemy Kibernet., 15, 117-134 (In Russian)
- [322] Spira, P.M: (1971a) On time-hardware complexity tradeoffs for Boolean functions; Proc. 4th Hawaii Int. Symposium on System Sciences, 525-527
- [323] Spira, P.M: (1971b) On the time necessary to compute switching functions; IEEE Trans. Computers, C-20, 104-105
- [324] Spira, P.M: (1973) Computation times of arithmetic and Boolean functions in (d, r) -circuits; IEEE Trans. Computers, C-22, 552-555
- [325] Stockmeyer, L: (1977) On the combinational complexity of certain symmetric Boolean functions; Math. Syst. Theory, 10, 323-336
- [326] Subbotovskaya, B.A: (1961) Realisations of linear functions by formulas using $\{\wedge, \vee, \neg\}$; Doklady Akademii Nauk SSSR, 136, 553-555 (In Russian) (Transl: Sov. Math.-Doklady, 2, 110-112)
- [327] Subbotovskaya, B.A: (1963) Comparison of bases in the realisation by formulas of functions of the algebra of logic; Doklady Akademii Nauk SSSR, 149, 784-787 (In Russian) (Transl: Sov. Math.-Doklady, 4, 478-481)
- [328] Tardos, E: (1988) The gap between monotone and non-monotone circuit complexity is exponential; To appear *Combinatorica*
- [329] Tarjan, R.E: (1978) Complexity of monotone networks for computing conjunctions; Ann. Discrete Math., 2, 121-133
- [330] Thompson, C.D: (1979) Area-Time complexity for VLSI; Proc. 11th ACM Symposium on Theory of Computing, 81-88
- [331] Thompson, C.D: (1980) A complexity theory for VLSI; Ph.D Dissertation; Report No. CMU-CS-80-140, Dept. of Computer Science, Carnegie-Mellon Univ.
- [332] Tiekhenrich, J: (1984) A $4n$ lower bound on the monotone Boolean network complexity of a one output Boolean function; Inf. Proc. Letters, 18, 201-202
- [333] Tkachev, G.A: (1980) On the complexity of a sequence of Boolean functions by implementing in terms of circuits and Π -circuits under additional restrictions on the circuits structure; Combinatorial-Algebraic methods in Applied Mathematics, Gorki, 261-267 (In Russian)
- [334] Toom, A.L: (1963) The complexity of a scheme of functional elements realising the multiplication of integers; Doklady Akademii Nauk, 150, 496-498 (In Russian) (Transl: Sov. Math.-Doklady, 3, 714-716)
- [335] Toom, A.L: (1967) Complexity of realisation of binary functions with small sub-functions; Problemy Kibernet., 18, 83-90 (In Russian) (Transl: Syst. Theory Res., 18 (1968), 77-84)

- [336] Trakhtenbrot, B.A: (1959) Asymptotic evaluation of the complexity of logic nets with memory; Doklady Akademii Nauk SSSR, 127, 281-284 (In Russian) (Transl: Autom. Expr., 2, 13-14)
- [337] Turing, A.M: (1936) On computable numbers, with an application to the Entscheidungsproblem; Proc. London Mathl. Soc. Series 2, 42, 230-265; Corrections: ibid 43, (1937), 544-546
- [338] Ugolnikov, A.B: (1976) Realization of monotonic functions by networks of functional elements; Problemy Kibernet., 31, 167-85 (In Russian)
- [339] Ugolnikov, A.B: (1979) The synthesis of schemes and formulas in incomplete bases; Sov. Math.-Doklady, 20, 1224-1227
- [340] Ugolnikov, A.B: (1983) On the realisation of functions from closed classes by schemes of functional elements in a complete basis; ibid, 28, 45-46
- [341] Ugolnikov, A.B: (1987) On the complexity of realising Boolean functions by schemes over the basis of majority and implication; Vestn. Mosc. Un-ta. Ser. 1, Matematika Mechanika, 4, 76-78 (In Russian)
- [342] Ulig, D: (1974) On the synthesis of self-correcting schemes from function elements with a small number of reliable elements; Matem. Zametki, 6, 937-944 (In Russian) (Transl: Math. Notes Acad. Sci. USSR, 15, 558-562)
- [343] Ullman, J.D: (1984) The complexity of VLSI algorithms; Addison-Wesley
- [344] Valiant, L.G: (1976) Universal circuits; Proc. of 8th ACM Symposium on Theory of Computing, 196-203
- [345] Valiant, L.G: (1979a) The complexity of computing the permanent; Theoretical Computer Science, 8, 189-201
- [346] Valiant, L.G: (1979b) Negation can be exponentially powerful; Proc. 11th ACM Symposium on Theory of Computing, 189-196
- [347] Valiant, L.G: (1979c) Completeness classes in algebra; Proc. 11th ACM Symposium on Theory of Computing, 249-261
- [348] Valiant, L.G: (1983) Exponential lower bounds for restricted monotone circuits, Proc. 15th ACM Symposium on Theory of Computing, 110-117
- [349] Valiant, L.G: (1984) Short monotone formulae for the majority function; Jnl. of Algorithms, 5, 363-366
- [350] Valiant, L.G: (1986) Negation is powerless for Boolean slice functions; SIAM Jnl. on Computing, 15, 531-535
- [351] Van Leijenhorst, D.C: (1987) A note on the formula size of the "*mod k*" functions; Inf. Proc. Letters, 24, 223-224
- [352] Van Voorhis, C.C: (1972) An improved lower bound for sorting networks; IEEE Trans. Computers, C-21, 612-613
- [353] Vaschenko, V.P: (1979) On the computation of all non-trivial simple decompositions of a function of the algebra of logic; Sov. Math.-Doklady, 20, 629-632
- [354] Vilfan, B: (1976) Lower bounds for the size of expressions for certain functions in d -ary logic; Theoretical Computer Science, 2, 246-69
- [355] Voigt, B; Wegener, I: (1988) Minimal polynomials for the conjunction of functions on disjoint variables can be very simple; Internal Report, Univ. of Dortmund, Nr. 252

- [356] Vuillemin, J: (1980) A combinatorial limit to the computing power of VLSI circuits; Proc. 12th ACM Symposium on Theory of Computing, 294-300
- [357] Wallace, C.S: (1964) A suggestion for a fast multiplier; IEEE Trans. Electron. Computers, EC-13, 14-17
- [358] Ward, M: (1946) Note on the order of free distributive lattices; Abstract 135, Bull. Amer. Math. Soc., 52, 423
- [359] Wechsung, G: (1977) A nonlinear lower bound for the formula complexity of certain Boolean functions; Information Processing 77, 831-833
- [360] Wegener, I: (1979) A counterexample to a conjecture of Schnorr referring to monotone networks; Theoretical Computer Science, 9, 147-150
- [361] Wegener, I: (1980) A new lower bound on the monotone network complexity of Boolean sums; Acta Informatica, 13, 109-114
- [362] Wegener, I: (1981) An improved complexity hierarchy on the depth of Boolean functions; Acta Informatica, 15, 147-152
- [363] Wegener, I: (1982) Boolean functions whose monotone complexity is of size $\frac{n^2}{\log_2 n}$; Theoretical Computer Science, 21, 213-224
- [364] Wegener, I: (1985) On the complexity of slice functions; Theoretical Computer Science, 38, 55-68
- [365] Wegener, I: (1986) More on the complexity of slice functions; Theoretical Computer Science, 43, 201-211
- [366] Wegener, I: (1987) The complexity of Boolean functions, Wiley-Teubner
- [367] Weiss, J: (1983) An $\Omega(n^{3/2})$ lower bound on the complexity of Boolean convolution; Information and Control; 59, 84-88
- [368] Winograd, S: (1965) On the time required to perform addition; Jnl. of the ACM, 12, 277-285
- [369] Winograd, S: (1967) On the time required to perform multiplication; Jnl. of the ACM, 14, 793-802
- [370] Yablonskii, C.V: (1954) Realisation of linear functions in the class of Π -schemes; Doklady Akademii Nauk SSSR, 94, 165-179 (In Russian)
- [371] Yablonskii, C.V: (1959a) On the impossibility of eliminating the trials of all functions in P_2 in solving certain problems in the theory of networks; Doklady Akademii Nauk SSSR, 124, 44-47 (In Russian)
- [372] Yablonskii, C.V: (1959b) On algorithmic obstacles in synthesis of minimal contact schemes; Problemy Kibernet., 2, 75-121 (In Russian)
- [373] Yamamoto, K: (1954) Logarithmic order of free distributive lattice; Jnl. Math. Soc. of Japan, 6, 343-353
- [374] Yao, A.C-C: (1983) Lower bounds by probabilistic arguments; Proc 24th IEEE Symp. on FOCS, 420-428
- [375] Yao, A.C-C: (1985) Separating the polynomial-time hierarchy by oracles: Part I; Proc. 26th IEEE Symp. on FOCS, 1-10
- [376] Yao, A.C-C; Yao, F.F: (1976) Lower bounds on merging networks; Jnl. of the ACM, 23, 566-571

- [377] Young, M.H; Muroga, S: (1985) Symmetric minimal covering problems and minimal PLAs with symmetric variables; IEEE Trans. Comput., C-34, 523-541
- [378] Zakharova, E.Y: (1972) The realisation of functions of the P_k by formulae ($k \geq 3$); Mat. Zametki, 11, 99-108 (In Russian)
- [379] Zhegalkin, I.I: (1927) The technique of calculation of statements in symbolic logic; Matem. Sbornik, 34, 9-28 (In Russian)
- [380] Zhuravlev, Y.I: (1979) Local algorithms over disjunctive normal forms; Sov. Math.-Doklady, 20, 286-289
- [381] Zhuravlev, Y.I; Kogan, A.Y: (1985) Realization of Boolean functions with a small number of zeros by disjunctive normal forms and related problems; *ibid*, 32, 771-775

Summary of Notations

The following lists summarise notation which is in use throughout most of the text above. Notation which is specific to only one chapter is not included.

1) Boolean functions and sets of functions

B_n	Set of all n -input single output Boolean functions
$B_{n,m}$	Set of all n -input, m -output Boolean functions
M_n	Set of all n -input single output monotone Boolean functions
$M_{n,m}$	Set of all n -input m -output monotone Boolean functions
S_n	Set of all n -input symmetric Boolean functions
\tilde{f}	Dual function of $f \in B_n$
$f ^\pi$	Subfunction of f induced by partial assignment π
\bar{f} or $\neg f$	Negation (complement) of $f \in B_n$
$[f_n]$	Family of Boolean functions
$f^{(n)}$	n 'th member of family $[f_n]$
π_1	Projection function $\pi_1(x, y) = x$
π_2	Projection function $\pi_2(x, y) = y$

\neg	Logical not (negation or complement)
\wedge	Conjunction
\vee	Disjunction
\Rightarrow	Left implication
\Leftarrow	Right implication
\Leftrightarrow	Equivalence
\oplus	Exclusive or
g^e	$g \oplus e \oplus 1; e \in \{0, 1\}, f \in B_n$
<i>ADD</i>	Integer addition
<i>BMP</i>	Boolean Matrix Product
C_k^n	Congruent mod k
<i>COMP</i>	Comparison
<i>CONV</i>	Boolean convolution
<i>DHC</i>	Directed Hamiltonian cycle
<i>DIVN</i>	Integer division
E_k^n	Exactly k
<i>MULT</i>	Integer multiplication
<i>PM</i>	Perfect matching (=Logical permanent)
<i>SAT</i>	Satisfiability
T_k^n	Threshold k
<i>UHC</i>	Undirected Hamiltonian cycle
\mathbf{X}_n	$\langle x_1, \dots, x_n \rangle$, ordered set of n Boolean variables
α	Assignment, $\alpha = \langle a_1, \dots, a_n \rangle \in \{0, 1\}^n$
γ_α	$\bigwedge_{\{i: a_i=1\}} x_i, \alpha \in \{0, 1\}^n$

$$\delta_\alpha \quad \bigwedge_{i=1}^n (x_i \iff a_i)$$

2) Complexity relations and classes

C	Combinational network complexity (of network or function)
C_Ω	Ω-network complexity
C^m	Monotone network complexity
D	Combinational network depth
D_Ω	Ω-network depth
L	Formula size (over basis B_2)
L_Ω	Ω-formula size
L^m	Monotone formula size
P	Deterministic polynomial-time computable
NP	Non-deterministic polynomial-time computable

When f, g are functions $\mathbf{N} \rightarrow \mathbf{N}$.

$f(n) = O(g(n))$ if and only if there is a constant $c > 0$ such that for all n , $f(n) \leq c \cdot g(n)$.

$f(n) = \Omega(g(n))$ if and only if $g(n) = O(f(n))$.

$f(n) = o(g(n))$ if and only if $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 0$

$f(n) = \omega(g(n))$ if and only if $g(n) = o(f(n))$.

3) Miscellaneous

CNF	Conjunctive Normal Form
DNF	Disjunctive Normal Form
$\mathbf{I}(f)$	Set of implicants of $f \in B_n$
$op(g)$	Operation associated with gate g in a network
$\mathbf{PC}(f)$	Set of prime clauses of $f \in B_n$
$\mathbf{PI}(f)$	Set of prime implicants of $f \in B_n$
$res(u)$	Boolean function computed by node u of a network
$var(h)$	Set of variables defining a monom or clause h
$\phi(u)$	Fanout of node u in a network
Ω	Arbitrary logical basis
$f \leq g$	$f(\alpha) = 1 \Leftrightarrow g(\alpha) = 1$ for all $\alpha \in \{0, 1\}^n$
2^S	Set of all subsets of a (finite) set S
$\lceil x \rceil$	Smallest integer which is $\geq x$
$\lfloor x \rfloor$	Largest integer which is $\leq x$

Bibliographic Categories

The following index is intended to summarise the bibliographic entries which are relevant to various specific areas within the field of Boolean complexity theory. The numbers next to headings refer to the bibliography ordering and are not page numbers.

Bounded-depth networks	3, 4, 35, 48, 51, 64, 80, 89, 110, 118, 155, 157, 175, 177, 184, 211, 234, 237, 315, 320, 333, 375
- $\{\wedge, \oplus\}$	239, 268, 269
- arbitrary basis	50, 117
- DNF	10, 11, 15, 16, 53, 54, 128, 158, 190, 192, 193, 229, 243, 263, 264, 281, 288, 355, 377, 380, 381
- monotone	36, 348, 374
- ringsum expansion	30, 79, 86, 121, 213, 236, 287, 296, 379
Combinational networks and complexity	44, 75, 186, 191, 219, 220, 225, 230, 231, 235, 241, 249, 257, 273, 285, 290, 291, 301, 303, 308, 314, 336, 344
- Arithmetic functions	9, 18, 24, 45, 126, 136, 160, 289, 304, 316, 317,

	334, 335, 357, 368, 369
- Disjoint variable sets	244, 342
- Hierarchies	197, 242, 362
- Lower bounds	17, 32, 106, 107, 129, 245, 276, 297, 299, 302, 309, 325
- Ω -networks	34, 156, 210
- Relationships to other models	38, 40, 84, 298, 313
- Restricted fan-out	114, 149
- Synthesis	52, 94, 95, 99, 102, 171, 176, 179, 181, 183, 216, 221, 222, 223, 224, 272, 273, 275, 312, 321, 339, 340
- Upper bounds	120, 135, 136, 159, 171, 215, 292
 Combinatorics and graph theory	 25, 47, 77, 78, 115, 144, 168, 265
 Complexity theory	 2, 58, 59, 85, 92, 111, 148, 208, 232, 233, 295, 337, 345
- abstract	20, 21, 116, 186
- algebraic	39, 122, 165, 166, 300, 346, 347
- Boolean	41, 293, 319, 366
- Parallel	61, 91, 97
- uniform	60, 74, 251, 286

Depth	9, 22, 29, 41, 62, 76, 93, 125, 141, 182, 194, 195, 198, 202, 214, 256, 258, 259, 307, 322, 323, 324
- Monotone	42, 43, 127, 196
Formulae	131, 132, 140, 154, 172, 180, 212, 217, 218, 254, 262, 277, 318, 327, 371, 372, 378
- Lower bounds	13, 19, 49, 56, 57, 83, 108, 112, 113, 137, 138, 139, 146, 147, 187, 189, 205, 226, 260, 261, 271, 282, 305, 326, 354, 359
- Monotone	49, 56, 88, 142, 178, 271, 278, 279, 349
- Synthesis	65, 81, 173, 174
- Unate	13, 57, 133, 134, 137, 138, 187, 261, 271, 326, 370
- Upper bounds	88, 133, 134, 142, 173, 174, 246, 247, 349, 351
Monotone Networks	31, 66, 164, 283, 306
- Disjoint variable sets	90
- Lower bounds	8, 12, 14, 33, 67, 69, 101, 123, 161, 162, 170,

	206, 207, 227, 228, 238, 252, 255, 266, 267, 270, 328, 329, 332, 352, 360, 361, 363, 367, 376
- Multivalued	6, 7
- Number of monotone functions	6, 55, 63, 98, 100, 103, 104, 105, 143, 144, 145, 150, 151, 358, 373
- Planar	28, 201
- Relationships to other bases	26, 70, 71, 72, 73, 82, 270, 328, 341, 350, 364, 365
- Replacement rules	27, 68, 207, 238
- Sorting	5, 23, 161, 240, 352
- Upper bounds	5, 23, 119, 240, 248, 250, 338
Planar Networks	169, 188, 199, 200, 203, 294
Switching theory	1, 37, 87, 96, 109, 130, 153, 163, 209, 253, 280, 310, 311, 353
VLSI	46, 152, 167, 204, 330, 331, 343, 356

Author Index

Abullaev, D.A.	422
Aho, A.V.	25, 422
Ajtai, M.	269, 337, 422
Alekseev, V.B.	268, 422, 423
Alon, N.	120, 195, 196-224, 232, 238, 269, 423
Alt, H.	115, 423
Andreev, A.E.	121, 195, 224-232, 238, 269, 272, 343, 347-351, 394, 423
Arevalo, Z.	394, 423
Aslanjan, L.A.	424
Avgustinovich, S.V.	424
Avizienis, A.	116, 424
Babai, L.	351, 424
Baker, T.	355-356, 424
Barak, A.B.	70, 424, 448
Batcher, K.E.	245, 424
Beame, P.W.	115, 424
Berge, C.	254, 297, 424
Berman, L.	451
Ben-Or, M.	422
Berkowitz, S.	121, 238, 239, 243-244, 248-249, 425
Beynon, W.M.	268, 421, 425
Bini, D.	425
Bioul, G.	394, 425
Bloniarz, P.	184, 352, 425

Blum, N.	74, 76, 90-99, 116, 161, 425
Boppana, R.	121, 195, 196-224, 232, 238, 269, 394, 423, 425, 426
Born, R.C.	426
Borodin, A.	25, 26, 37-39, 426
Bredeson, J.G.	394, 423
Breitbart, Y.Y.	115, 426
Bremer, H.	116, 426
Brent, R.	68-69, 413, 414, 426
Brown, W.G.	170, 427
Brustmann, B.	377, 427
Bublitz, S.	351, 427
Buckle, J.	421, 425
Chandra, A.K.	373-377, 395, 427
Chiang, K-W.	427
Chuang, Y.H.	394, 434
Chukhrov, I.P.	394, 427
Church, R.	123, 427
Commentz-Walter, B.	272, 323-330, 428
Conway, L.	396, 443
Cook, S.A.	6, 25, 42, 115, 424, 428
Davio, M.	394, 425
Dedekind, R.	118, 119, 123, 428
Denenberg, L.	428
Deschamps, J.P.	394, 425
Dudich, V.N.	428
Dunne, P.E.	120, 121, 149-154, 172-191, 239, 242-243,

	249, 251, 252-253, 253-259, 260, 261-263, 265-268, 269, 429
Edenbrandt, A.	429
Ehrenfeucht, A.	429
Elspas, B.	58, 430
Erdős, P.	209, 297, 430
Even, S.	394, 430
Fagin, R.	377-378, 430
Finikov, B.I.	274-277, 430
Fischer, M.	25, 27-28, 32-36 100, 101-111, 269, 270, 272, 280, 303-322, 331, 430, 431, 438
Fleisher, H.	394, 431
Fortune, S.	395, 427
Friedman, A.D.	25, 431
Friedman, J.	339-343, 431
Furst, M.	354-356, 372, 377, 431
Gal, S.	426
Galbiati, G.	270, 431
Galil, Z.	120, 148-150, 157-160, 269, 431, 443
Garey, M.	25, 431
Gaskov, S.B.	43, 51, 58-63, 431
Gavrilov, M.A.	431
Gershkovich, Y.B.	432
Gibbons, A.M.	413, 432
Gilbert, E.N.	123, 432

Gill, J.	355-356, 424
Gimpel, J.F.	14, 394, 432
Greene, C.	125, 432
Grigoriev, D.Y.	432
Gurevich, I.B.	432
Gurevich, Y.	428
Hansel, G.	119, 123-133, 136, 432
Harper, L.H.	116, 272, 280, 288-289, 433
Harrison, M.A.	25, 433
Hastad, J.	356, 364-372, 377, 381, 433
Hennie, F.C.	32, 433
Hodes, L.	272, 280, 293-303, 351, 433
Hoover, H.J.	115, 424, 428, 433
Hopcroft, J.E.	25, 224, 356, 422, 433, 434
Hromkovic, J.	395, 434
Hsieh, W.N.	116, 433
Huynh, D.T.	434
Hyafil, L.	434
Jagadeesan, M.	394, 434
Jerrum, M.	119, 434
Johnson, D.	25, 431
Jukna, S.P.	269, 434
Jung, H.	115, 434
Karatsuba, A.	116, 434
Karchmer, M.	352, 435
Karnaugh, M.	14, 394, 435

Karp, R.M.	224, 433
Karpova, N.A.	115, 435
Karpovski, M.G.	435
Kautz, H.W.	58, 430
Kasim-Zade, O.M.	435
Khasin, L.S.	337-339, 435
Khazotskii, V.E.	431
Khrapchenko, V.M.	111, 115, 272, 337, 343, 344-347, 435, 436
Klawe, M.	377-378, 430, 433
Kleiman, M.	339, 436
Kleitman, D.	119, 123-124, 125, 140, 432, 436, 437
Kloss, B.M.	289-290, 437
Knuth, D.	58, 308, 437
Kodanpani, K.L.	437
Kogan, A.Y.	394, 460
Kohavi, I.	394, 430
Komlos, J.	269, 337, 422
Korobkov, V.K.	123, 437
Korshunov, A.D.	68, 119, 124, 147, 437
Kramer, M.	421, 437
Krichevskii, R.E.	331, 337, 437, 438
Kriegel, K.	438
Kuck, D.J.	68-69, 427
Kung, H.T.	413, 414, 427, 434
Kuznetsov, O.P.	431
Kuznetsov, S.E.	394, 438
Ladner, R.E.	100, 101-111, 438
Lagarias, J.C.	425

Lai, H.C.	438, 444
Lamagna, E.A.	161, 269, 438
Langheld, E.	439
Lenz, K.	270, 439
Levey, S.Y.	439
Lingas, A.	119, 439
Lipton, R.	395, 399-401, 413, 415, 418, 420, 427, 439
Long, D.	269, 439
Lupanov, O.B.	42, 45-50, 51, 58, 116, 119, 136-138, 274, 278-280, 354, 356, 357, 359-360, 360-363, 439-441
Machtley, M.	25, 441
Madatjan, H.A.	441
Malyshev, V.A.	351, 437, 441
Mamatov, Y.A.	290, 394, 420, 441, 442
Markov, A.A.	269, 442
Markowsky, G.	119, 123-124, 140, 437
Maruyama, K.	68-69, 427
Masek, W.J.	14, 442
McCluskey, E.J.	14, 394, 442
McColl, W.F.	43, 51-58, 64-65, 70, 245, 268, 339, 352, 397, 403, 404-405, 406-410, 410-412, 420, 421, 442, 443
McKinney, M.H.	394, 451
Moskalev, E.S.	435
Mead, C.	396, 443
Mehlhorn, K.	120, 148-150, 157-160, 166-170, 269, 351, 443

Meyer, A.	25, 272, 280, 303-322, 331, 354, 430, 443
Miller, R.E.	25, 443
Mirwald, R.	270, 443
Moran, S.	377-380, 443
Muchnik, B.A.	443
Mukhopadhyay, A.	394, 444
Muller, D.E.	58, 70, 100, 111-114, 115, 444, 448
Munro, I.	6, 426
Muroga, S.	394, 438, 444, 460
Nakasima, A.	271, 444
Neciporuk, E.I.	166, 272, 280, 281-293, 347, 352, 444-445
Nguen, K.A.	394, 445
Nigmatullin, R.G.	116, 445
Noe, P.S.	394, 451
Ofman, Y.	116, 434, 446
Okol'nishnikova, B.	381, 446
Ong, E.H.	394, 451
Orlov, V.A.	115, 446
Pan, V. Ya.	425
Papakonstantinou, G.	394, 446
Parberry, I.	395, 446
Paterson, M.S.	27, 43, 51-58, 65-67, 68, 70-73, 120, 148, 156-160, 245, 269, 272, 280, 282-284, 303-322, 331, 344-346, 383-393, 397, 412, 430, 443, 446-447
Paz, A.	394, 430

Paul, W.	75, 76, 77, 90, 93, 116, 352, 431, 447
Paull, M.C.	439
Peterson, G.L.	331, 447
Pippenger, N.J.	25, 27-28, 32-36, 116, 136, 139-140, 161, 269, 339, 377-378, 430, 433, 436, 447-448
Poltervich, V.M.	432
Pooch, U.W.	394, 451
Post, E.L.	11-12, 448
Pratt, V.R.	269, 323, 448
Preparata, F.P.	58, 70, 100, 111-114, 115, 444, 448
Pudlak, P.	272, 280, 293-303, 331, 351, 424, 448
Pulatov, A.K.	449
Pupyrev, E.I.	449
Quine, W.V.	14, 394, 449
Rabin, M.O.	25, 430
Ramsey F.P.	296-298, 449
Razborov, A.A.	120, 195, 196-224, 232-233, 269, 352, 383-393, 395, 449-450
Red'kin, N.P.	116, 119, 140-146, 352, 450
Reiter, B.	426
Reznik, V.I.	450
Reischer, C.	451
Rhyne, T.V.	394, 451
Riordan, J.	50, 271, 273-274, 451
Rivest, R.L.	270, 451
Rodl, V.	351, 424
Romankevich, H.M.	451

Rudich, S.	451
Ruzzo, W.L.	25, 451
Rytter, W.	413, 432
Saluja, K.K.	394, 451
Sapozhenko, A.A.	394, 451
Sarkisjan, G.Z.	451
Sattler, J.	272, 323, 330, 428
Savage, J.E.	116, 261, 269, 272, 280, 288-289, 410, 413, 414-415, 416-418, 420, 433, 438, 452
Savitch, W.J.	7, 452
Saxe, J.B.	354-356, 372, 377, 431
Schmitz, G.	394, 444
Schmookler, M.S.	394, 452
Schnitger, G.	395, 446
Schnorr, C.P.	28, 29-32, 74, 76, 77-79, 116, 119, 269, 270, 443, 452-453
Schönhage, A.	115, 453
Schürfeld, U.	272, 280, 290-293, 453
Scidmore, A.K.	426
Sedgewick, R.E.	413, 439
Selman, A.	356, 424
Seth, S.C.	437
Sethi, I.K.	453
Seysen, M.	116, 425
Shamir, E.	70, 424, 453
Shannon, C.E.	27, 42-45, 50, 119, 271, 273-274, 405, 451, 453
Shelah, S.	428

Shestakov, V.I.	271, 453
Sholomov, L.A.	115, 116, 453-454
Simovici, D.	451
Sipser, M.	354-356, 372, 377, 381, 431, 454
Sklansky, J.	116, 454
Skyum, S.	27, 40-42, 269, 373, 454
Smolensky, R.	395, 454
Snir, M.	119, 434, 453
Solovay, R.	355-356, 424
Soprunenko, E.P.	454
Specker, E.	272, 280, 293-303, 433
Spencer, J.	209, 297, 430
Spira, P.M.	52, 70, 116, 323, 455
Stearns, R.E.	32, 433
Stockmeyer, L.	25, 74, 76, 77-89, 184, 354, 373-377, 377-378, 427, 430, 443, 455
Stone, H.W.	58, 430
Strassen, V.	115, 453
Subbotovskaya, B.A.	343, 347, 351, 455
Szemerédi, E.	269, 337, 351, 422, 424
Tardos, E.	224, 455
Tarjan, R.E.	269, 399-401, 414, 415, 418, 420, 439, 455
Tavel, M.	394, 431
Thompson, C.D.	413, 455-456
Tiekenheinrich, J.	120, 172, 191-192, 456
Tkachev, G.A.	356, 456
Toom, A.L.	116, 456
Trakhtenbrot, B.A.	456

Turing, A.M.	2, 456
Ugolnikov, A.B.	140, 239, 263, 265-267, 456-457
Ulig, D.	116, 457
Ullman, J.D.	25, 26, 356, 396, 422, 434, 457
Valiant, L.G.	27, 40-42, 68, 70-73, 115, 161, 239, 244-248, 269, 333-337, 373, 394, 447, 448, 454, 457-458
Van Leijenhorst, D.C.	333, 458
Van Leeuwen, J.	421, 437
Van Voorhis, C.C.	269, 458
Vascenko, V.P.	458
Vilfan, B.	280, 294, 458
Vishkin, U.	373-377, 427
Voigt, B.	458
Vranesic, Z.G.	427
Vuillemin, J.	413, 421, 458
Waack, S.	438
Wallace, C.S.	116, 458
Ward, M.	123, 458
Wechsung, G.	351, 458
Wegener, I.	65-67, 121, 165, 166, 170-171, 224, 232, 239, 244, 249, 251-252, 258, 259-260, 261, 269, 270, 377, 427, 439, 447, 458-459
Weiss, J.	120, 161-164, 459
Wigderson, A.	352, 435
Winograd, S.	459

Yablonskii, C.V.	116, 459-460
Yamamoto, K.	460
Yao, A.C-C.	356, 364, 372, 394, 460
Yao, F.F.	460
Yatsunov, A.I.	451
Yeager, J.D.	394, 431
Young, M.H.	394, 460
Young, P.	25, 441
Yunosov, D.	422
Zakharova, E.Y.	460
Zhegalkin, I.I.	14, 460
Zhuravlev, Y.I.	394, 432, 460

Subject Index

Absorption property	10
addition	18, 107-111, 115, 116, 374
affine	11
algorithm	2
almost all	43
ancestor	19
AND (\wedge)	8
\wedge -type function	20
Area-Time complexity	413-5
arithmetic functions	18, 115, 116
assignment	9
associative property	10
Basis	11, 18
bipartite graph	170
Boolean algebra	7-18
Boolean function	7-18
Boolean matrix product	120, 149, 155-160
Boolean (Ω)-network	18-25
Boolean sum	12, 165
Boolean product	12, 45
Boolean variable	7
bottom level fanin	370
bound pair	125
boundary	377-378

bounded alternation	357
bounded-depth formula	357
bounded-depth network	117, 353-395
- $\{\wedge, \vee, \neg\}$	353-382
- Upper bound (all functions)	359-363
- Lower bounds	364-381
- $\{\wedge, \oplus, 1\}$	382-394
Canonical slice function	251-253
central slice function	251, 253-258
centre (of sphere)	58
chain	124
clause	15
clique function	195, 213-218, 250, 252, 254-255, 290-293
<i>CLOSED</i> (f)	197, 200-208, 209-213
collector	94
combinational complexity	20, 27-116
- and Depth	39, 68, 70-73
- and Formula size	39
- Lower bound	36, 43-45, 74-99
- Upper bound	45-50, 100-115
combinational network	19, 27-116, 117
commutative property	10
comparison function	374
complement	9, 11
complement property	10

complete (logical) basis	11, 22
complexity class	5, 24
complexity gap	63-68
complexity hierarchy	5, 63-68, 120, 146-7, 381-382
complexity measure	4, 20, 39
complexity theory	1
congruent mod k (C_k^n)	17, 89, 322, 331-333
conjunction	8
conjunctive expansion	51
conjunctive normal form (CNF)	12-13, 15, 357, 359-360
constant-depth, polynomial-size	373-382
constant-depth reduction	373
constant function	8
constant property	10
convolution	119, 160-164
counting argument	42-45
cover (of basis)	11
cyclic convolution	160-164
De Morgan's Laws	10, 240, 344, 383
decision problem	1, 7, 24
Dedekind's problem	118-119, 122-133
degenerate Boolean function	9, 65-66
depth	20, 36-40, 50-64
- Lower bound	50
- Upper bound (schemes)	52-58

- Upper bound (networks)	58-63
depth-universal	115
descendant	19
determinant (<i>DET</i>)	289-290
deterministic	4
Direct Matrix Product (<i>DMP</i>)	171-172
disjunction	8
disjunctive expansion	51, 72
disjunctive normal form (DNF)	12-13, 15, 357, 359-360
distributive property	10
division	115
<i>DLOGSPACE</i>	6, 42
<i>DSPACE</i>	5
<i>DTIME</i>	5, 24, 36
dual function	11
embedding	397-398
equivalence (\Leftrightarrow)	8
equivalence function (δ_α)	46-47
exactly k (E_k^n)	17
exclusive-or (\oplus)	8
\oplus -type function	20
explicitly defined	74
expression	11
Family (of functions)	24, 75
fanin	18, 22
fanout	18, 23

finite state transducer	104
first node	155
formula	23-24, 117, 271-352
formula size	24, 271-352
- and depth	39, 68-70, 322-330
- Lower bound (almost all functions)	50, 272, 273-274, 280-322
- Lower bounds (\wedge, \vee, \neg)	343-351
- Lower bounds (B_2)	281-322
- Upper bound (all functions)	274-280
- Upper bounds (B_2)	330-333
- Upper bounds (monotone)	334-343
free input functions	165, 224, 232
free path	94
free split	94
free 0	125
free 1	125
Gate	18
graph	18
graph theory	117
Hamiltonian circuit	250, 252, 255-257, 355
Hamming distance	341
(h, k) -disjoint set of sums	165-170
homogeneous formula	295

Idempotency	10
identity	10
implicand	12
implicant	12
incomplete (logical) basis	11
inductive gate elimination	75, 116, 147, 154, 172
input	18
isomorphic formulae	295
<i>k</i> -formula	296
<i>k</i> -sensitive function	302-303, 306, 321
Language	4
last node	155
lattice method	196-224, 232
left implication (\leftarrow)	8
level	20
lexicographic ordering	45
local coding	119, 136-140
literal	11
logical basis	11
<i>LOGSPACE</i> -complete	6
Lupanov decomposition	46-49, 278-280
Majority (MAJ_n)	17, 89, 184-191, 269, 306, 333-337, 343, 393
marriage problem	280

merging network	245
model of computation	2
monom	15
monotone	15
monotone basis	16, 118
monotone Boolean function	15-17
monotone Boolean network	19, 117-270
monotone network complexity	117-270
- and combinational complexity	238-268
- Lower bound (almost all)	122
- Lower bounds (sets)	154-172
- Lower bounds (single output, linear)	172-192
- Lower bounds (single output, superlinear)	195-238
- Upper bound (combinational networks)	136-140
- Upper bound (monotone networks)	140-146, 192-194
monotone formula	
monotone projection	40
multiplication	115, 116, 119, 374, 415, 420
<i>n</i> -ordered network	46
NAND ($\neg \wedge$)	8
negation (\neg)	9, 224, 269
network	18
network depth	20
network size	20
node	18
non-degenerate	9, 65-66
non-deterministic	4

non-uniform	24, 42
NOR ($\neg \vee$)	8
<i>NP</i>	5, 40
<i>NP</i> -complete	6, 25, 41, 118, 249
<i>NSPACE</i>	5, 36
<i>NTIME</i>	5
Oblivious	28
occurrences (of literal in formula)	282
optimal network	20
OR (\vee)	8
oracle	355-356
output	18
<i>P</i>	5, 354
<i>pC</i>	41
<i>pD</i>	41
<i>p</i> -complete	41
<i>p</i> -definable	40
<i>p</i> -projection	40
<i>p</i> -universality	40
parallel prefix	100-104
parity function	346, 356, 359-360, 364-372
partial assignment	9
perfect matching	195, 213, 218-224, 268
planar crossover	401, 404-405
planar monotone computation	421

planar network	117, 398
planar network complexity	396-421
- and combinational networks	397-405
- Lower bound (almost all functions)	405-410
- Lower bounds	416-420
- Upper bound	410-412
planar separator theorem	415
polynomially reducible	6
polynomial time	5
polynomial-time hierarchy	354-356
predecessor	18
<i>PREFIX</i>	273
prefix problem	100-104
prime clause	12, 15
prime clause extension	152
prime implicant	12, 15
prime implicant extension	152
principle of duality	11
probabilistic method	209, 364-372
problem size	2
product network	100-104
programmable logic array (PLA)	394
projection	40
projection functions (π_1 etc)	8
pseudo-complement	121, 239-243
<i>PSPACE</i>	7, 42, 354
Quadratic Boolean form	270

Ramsey property	296
random assignment	365
reducing assignment	365
regular lattice	196
relativisation	354-356
relay-contact network	271
replacement rule	120, 147-154, 164, 239
restricted model	117
restriction (of function)	294
right implication (\Rightarrow)	8
ringsum expansion	14, 270, 286-287
<i>RL</i> -specification	407
Satisfiability (<i>SAT</i>)	41, 250, 252-253, 257-258
satisfying assignment	9
scheme	51
self-dual	11
Shannon function	115
shifting convolution	160
shifting function	418
simulation (TM space by Depth)	36-39
simulation (TM time by Size)	28-36
size	20, 39
slice function	121, 239, 243-263
space complexity	4
spectrum	17, 377
sphere	58

split	94
Stable Marriage Problem	288-289
standard circuit	121, 239-243
strong-complete basis	11
subfunction	9
successor	18
switching theory	14
symmetric Boolean function	17, 77, 100, 112-114, 281, 293, 303, 330-343, 377-380
sympathetic basis	265
Threshold function (T_k^n)	17, 89, 118, 172-191, 192-194, 322, 330, 331, 337-343, 347
<i>TIED</i> (and properties)	126-130
time complexity	4
topological order	20
transitive closure	36, 374
truth-table	8-9
Turing machine	2
Uniform circuit complexity	24
universal function	40, 115
universal circuit	115
Variable	7

Weak-complete basis	11
well-formed string	125
wire	18
wire counting	172, 177-183
Value function	170-171
VLSI	26, 396, 413-415