This blank page was inserted to preserve pagination.

THE COMPLEXITY OF FINITE FUNCTIONS

ABSTRACT

Lower bounds on the length of formulas for finite functions are obtained from a generalization of a theorem of Specker. Let $f: \{0,1,\ldots,b-1\}^n \to \{0,1,\ldots,b-1\}$ be a function which can be represented by a formula of length $\leq c \cdot n$. Then f is a restriction $f': \{0,1,\ldots,b-1\}^m \to \{0,1,\ldots,b-1\}$ of a function which is representable by a special formula called nonpseudo-complexes. By showing that certain functions do not have restrictions representable by nonpseudo-complexes, we are able to conclude that the length of formulas representing the mod p sum, $p > 2$, or the connectedness of a pattern on a discrete retina cannot be bounded by a linear function of the number of variables in the formula.

Also considered are perceptrons over finite fields (cyclic perceptrons). It is shown that cyclic perceptrons of bounded order cannot represent the geometric predicate connectivity. An interesting aspect of this is that one proof of the corresponding result for bounded order perceptrons over the rationals rests on the inability of the latter to represent the parity function. However, the parity function requires order 1 if the field has characteristic 2. Thus, this proof breaks down in the case of cyclic perceptrons. Another geometric predicate that cannot be represented by bounded order cyclic perceptrons is Euler number equals k (for an arbitrary k). However, this predicate can be represented by bounded order perceptrons over the rationals. It must be noted, however, that our proofs are different and much simpler than the corresponding proofs derived by Minsky and Papert for perceptrons over the rationals.

Finally, we investigate k-pattern spectra of a discrete retina. This is the 2^{k^2}-tuple, each component of which corresponds to the number of times a particular kxk pattern occurs on the retina. It is shown that [illegible] determined from k-pattern spectra of discrete figures are functions of the Euler number of the figure.

Cambridge **Massachusetts 02139**

This report reproduces a thesis of the same title submitted to the Department of Electrical Engineering, Massachusetts Institute of Technology, in partial fulfillment of the requirements for the degree of Doctor of Philosophy, February 1972.

THE COMPLEXITY OF FINITE FUNCTIONS

ABSTRACT

Lower bounds on the length of formulas for finite functions are obtained from a generalization of a theorem of Specker. Let f: $\{0,1,\ldots,d-1\}^n \rightarrow \{0,1,\ldots,d-1\}$ be a function which can be represented by a formula of length $\leq c \cdot n$. For any m, if n is sufficiently large, there is a restriction f': $\{0,1,\ldots,d-1\}^m \rightarrow \{0,1,\ldots,d-1\}$ of f which is representable by a special class of formulas called homogeneous e-complexes. By showing that certain functions do not have restrictions representable by homogeneous e-complexes, we are able to conclude that the length of formulas representing the mod p sum, $p > d$, or the connectedness of a pattern on a discrete retina cannot be bounded by a linear function of the number of variables in the formula.

Also considered are perceptrons over finite fields (cyclic perceptrons). It is shown that cyclic perceptrons of bounded order cannot represent the geometric predicate connectivity. An interesting aspect of this is that one proof of the corresponding result for bounded order perceptrons over the rationals rests on the inability of the latter to represent the parity function. However, the parity function requires order 1 if the field has chracteristic 2; thus, this proof breaks down in the case of cyclic perceptrons. Another geometric predicate that cannot be represented by bounded order cyclic perceptrons is Euler number equals k (for an arbitrary k). However, this predicate can be represented by bounded order perceptrons over the rationals. It must be noted, however, that our proofs are different and much simpler than the corresponding proofs derived by Minsky and Papert for perceptrons over the rationals.

Finally, we investigate k-pattern spectra of a discrete retina. This is the 2^{k^2}-tuple, each component of which corresponds to the number of times a particular kxk pattern occurs on the retina. It is shown that the only topological predicates that can be determined from k-pattern spectra of discrete figures are functions of the Euler number of the figure.

This report reproduces a thesis of the same title submitted to the Department of Electrical Engineering, Massachusetts Institute of Technology, in partial fulfillment of the requirements for the degree of Doctor of Philosophy, February 1972.

ACKNOWLEDGEMENTS

In the first place, I owe a debt of gratitude to Professor Albert R. Meyer in the course of association with whom I learned the heuristics of research. In particular, he introduced me to the problems described here, and then spent long hours with me suggesting improvements and modifications.

I am also indebted to my readers, Professors Michael J. Fischer and C. L. Liu for valuable suggestions.

I would like to thank Professor Frederick C. Hennie and Project MAC for financial support during my study.

Last but not least, thanks are due to Miss Marsha Baker for consenting to type this thesis.

CONTENTS

CHAPTER ONE:	INTRODUCTION AND SURVEY	5
	1.1 Finite Functions	5
	1.2 Formulas	5
	1.3 Measures of Complexity	9
	1.4 Problems Related to the Length Measure	13
	1.5 Specker's Theorem	20
	1.6 Cyclic Perceptrons	23
CHAPTER TWO:	A GENERALIZATION OF A THEOREM OF SPECKER	26
	2.1 e-Complexes	26
	2.2 The Generalized Specker's Theorem	38
	2.3 On Specker's Theorem	41
CHAPTER THREE:	APPLICATIONS OF THE GENERALIZED SPECKER THEOREM	50
	3.1 Counting mod p	51
	3.2 Connectivity	55
	3.3 The Length of Symmetric Functions	60
CHAPTER FOUR:	CYCLIC PERCEPTRONS	74
CHAPTER FIVE:	PATTERN COUNTING MACHINES	89
APPENDIX A:	CERTAIN PROPERTIES OF SHORT FORMULAS	97
APPENDIX B:	THE LENGTH OF THE MOD 2 SUM OVER Π	112
LITERATURE		116
BIOGRAPHICAL NOTE		118

CHAPTER ONE

INTRODUCTION AND SURVEY

1.1 Finite Functions

Let n be nonzero and finite; then a partial function $\mathbb{N}^n \to \mathbb{N}$, defined on only finitely many n-tuples, is called a finite function. We will restrict our attention to a subclass of finite functions. $D = \{0,1,\ldots,d-1\}$ is an initial interval of $\mathbb{N}$. Then we will consider $\mathcal{F}$, the set of all (total) functions $D^n \to D$ for all possible D and (finite and nonzero) n.

Let $f: D^n \to D$. Then f is identified with a (functional) table with d^n rows (corresponding to all possible n-tuples over D) and n+1 columns (corresponding to the n arguments and the value of f). Obviously the number of functions $D^n \to D$ is d^{d^n}.

Consider any function $f: D^n \to D$ for arbitrary D, n. We will say that f depends on the ith argument if and only if there exist two n-tuples $\underline{a} = (a_1,\ldots,a_i,\ldots,a_n)$ and $\underline{b} = (b_1,\ldots,b_i,\ldots,b_n)$ such that $a_j = b_j$ for $j \neq i$, $a_i \neq b_i$, and $f(\underline{a}) \neq f(\underline{b})$. Suppose that f does not depend on its jth argument; then we will say that the jth argument is a fictitious argument.

1.2 Formulas

Let there be given the countable sets $\Xi = \{x_1,x_2,\ldots\}$ of variable symbols and Ω of operator symbols. Each element of Ω is a name for a function in $\mathcal{F}$, and conversely each function in $\mathcal{F}$ has a name in Ω. Let $\varphi \in \Omega$ represent the function $f: D^n \to D$. Then we will write $\arg(\varphi) = n$ and $\mathrm{dom}(\varphi) = D$.

1.2.1 Definition

A D-formula is a finite expression $F = \varphi(G_1,\ldots,G_n)$ such that $\varphi \in \Omega$, $\arg(\varphi) = n$, $\mathrm{dom}(\varphi) = D$, and either $G_i \in \Xi$ or G_i is a D-formula for $1 \leq i \leq n$.

A formula is simply a D-formula for some D.

Let F be an arbitrary D-formula and let x_n be the highest numbered variable symbol appearing in F. Then F represents a function $f: D^n \to D$. This correspondence is well-known and we will not describe it in detail. Without danger of imprecision, F will also be considered as a representation for all functions obtained from f by adding fictitious arguments.

Let there be given two formulas F and G. Suppose that F represents a certain function f, and also a representation for f can be obtained from G by possibly choosing different variable symbols. Then we will say that F is equivalent to G ($F \equiv G$).

Remarks. Usually, if we are dealing with D-formulas for a single domain D, we represent the identity function by a variable symbols (i.e., we omit the operator symbol for the identity). In the formal model we use, we cannot do this since it would be ambiguous. Also, for purely technical reasons, we insist that every operator has at least one argument (otherwise, the wording of several definitions and results would be more cumbersome). Thus, we do not allow constants. Rather, instead of constants, we use operators with one fictitious argument. Suppose we are given the formula F. Occasionally, we will say "Replace the variable x (in F) by the constant a". This is to be interpreted as "Replace the variable x with a(y)" where y is a variable symbol not appearing in F.

Let $f: D^n \to D$ and let g be an arbitrary finite function of n arguments with domain $E \subseteq D^n$ and such that $g = f \quad E$. Let F be a D-formula for (i.e., representing) f. Then we can also say that (F,E) represents g. From now on we will not be pedantic, and we will simply say that F represents g. Some of the main results in this thesis are concerned with the question, given a specific function $D^n \to D$, how much can we simplify its representation if we choose an E-formula for it with $D \subsetneq E$.

If F is an arbitrary formula, then the set of variables appearing in it will be called its <u>support</u> (denoted by S(F)). The set of operators appearing in F will be called its <u>basis</u> (denoted by B(F)).

Let $\Phi \subseteq \Omega$. Then the set of formulas F such that $B(F) \subseteq \Phi$ will be called the set of formulas <u>over</u> Φ. Hopefully without too much danger of ambiguity, we will also say that Φ is a basis of operators (for formulas over Φ). All the significant results we will describe deal with formulas over Φ when Φ is finite (and representing a set of operators with domain D for a single value of D). From now on, whenever a basis of operators Φ is introduced, it is always assumed finite. Usually, we are interested only in bases that allow all function $D^n \to D$ for a certain D and arbitrary n to be represented. Such bases will be called complete bases (for D).

Notation. Elements of $\mathcal{F}$ will always be denoted by lower case Latin letters. The various bases of operators we will use will be denoted by capital Greek letters; operators (i.e., basis elements) will be denoted by lower case Greek letters (except for well known operators for which established notation exists); formulas will be denoted by capital Latin letters; and D will always refer to the domain of formulas. d will denote $\overline{D}$.

1.2.2 Example.

If D = {0,1}, then the functions $D^n \to D$ for arbitrary n are known as Boolean functions. A complete basis for {0,1} conists of the binary operators $\wedge$ (conjunction) and $\vee$ (disjunction), and the unary operator $-$ (complementation). This basis shall be denoted by Π. The formula $F = \vee(\wedge(-(x_1),x_2),\wedge(x_1,-(x_2)))$ over Π represents $x_1 \oplus x_2$ (the mod 2 sum of x_1 and x_2). Usually, this is written as $\bar{x}_1 \wedge x_2 \vee x_1 \wedge \bar{x}_2$. We have $S(F) = \{x_1,x_2\}$, and $B(F) = \Pi$.

A convenient representation of formulas is by trees. This is a standard device that will not be described; suffice it to say that to each formula F there corresponds a tree T(F) whose terminal nodes are labelled with variable symbols and the nonterminal nodes with basis symbols. As an example, let F be as defined in Example 1.2.2. Then T(F) is shown in Fig. 1.1.

Given a formula F, we need a notation for subformulas of F.

The definition of <u>subformula</u> is the standard one: (1) F is a subformula of F, (2) if $F = \varphi(F_1,\ldots,F_k)$, then if F_i for $1 \le i \le k$ is not a variable symbol, any subformula of F_i is a subformula of F, and (3) subformulas of F are only objects satisfying (1) and (2). Subformulas distinct from F are proper subformulas.

Let G be a subformula of F such that $G = \varphi(H_1,\ldots,H_\ell)$. Then we will say $H_i = G_{.i}$ for $1 \le i \le \ell$. This notation can be iterated. In Example 1.2.2, $F_{.2.2} = -(x_2)$. However, note that $F_{.2.1}$ is a variable symbol which according to our definition is not a formula. This can be remedied by replacing this particular occurrence of the variable symbol x_1 by $id(x_1)$. For this reason we will require that all the bases we consider contain the identity function whether this is specifically mentioned or not.

If $G = F_{.j(1).j(2)...j(r)}$, then $j = j(1)j(2)...j(r)$ is called the index of G (for completeness, let λ denote the index of F). If G is a proper subformula of F, then $F = H(X,G)$ where $X \cup S(G) = S(F)$ and $H(X,z)$ is a formula (determined by j) where z appears only once. We write $H = F/G$. In this case, with F and G as given, we will also write $S^*_F(G) = X$ (i.e., the variables of F that appear outside of G). We define $S^*_F(F) = \phi$. The subscript F will generally be suppressed when it will be clear to what formula F we refer to. In what follows, whenever we will deal with a subformula G of F, it will be assumed that the index of G is also given; for if not, then, e.g., F/G and S*(G) are not uniquely defined.

Frequently, formulas will occur where certain variables have been replaced with constants. Suppose F is a formula over Φ, $X \subseteq S(F)$, and $a \in D$; then, F with all variables except those in X replaced by a will be denoted by F^X_a. Obviously, F^X_a is a formula over $\Phi \cup \{a\}$. If f is an arbitrary function, X a subset of its arguments, then f^X_a has the analogous meaning, viz., the function obtained from f by restricting the elements outside of X to a. The functional table of f^X_a is obtained from that of f by deleting all columns except those that correspond to X and retaining only the rows with a entries in the deleted columns.

1.3 Measures of Complexity

Let us introduce the three most widely studied measures on formulas:

(1) Length. The length of a formula F, denoted by L(F), is the number of occurrences of variable symbols in F. In other words, it is the number of terminal nodes of T(F).

(2) Cost. The cost of a formula F, denoted by C(F), is the number of operator symbols in F. In other words, it is the number of nonterminal nodes of T(F).

(3) Depth. The depth of a formula F, denoted by D(F), is the depth of nesting of operators in F. In other words, it is the number of arcs on the longest branch of T(F).

Now, given an arbitrary function $f: D^n \to D$ and a (finite) basis Φ, we define the length of f over Φ as

$$L(f,\Phi) = \min(\{\ell: \text{ There exists a formula F over } \Phi \text{ for f such that } L(F) = \ell\})$$

If f cannot be represented by a formula over Φ, we define $L(f,\Phi) = \infty$. Similarly, for cost and depth.

It is noteworthy that all the measures above are closely related. In fact,

$$c_0 \cdot L(f,\Phi) \leq c(f,\Phi) \leq c_1 \cdot L(f,\Phi) \tag{1.3.1}$$

$$c_2 \cdot \log_2(L(f,\Phi)) \leq D(f,\Phi) \leq c_3 \cdot \log_2(L(f,\Phi)) \tag{1.3.2}$$

for an arbitrary function f such that $L(f,\Phi)$, $C(f,\Phi)$, and $D(f,\Phi)$ are finite, and certain constants c_0, c_1, c_2, and c_3 that depends on Φ. The basis Φ is also arbitrary, except in the case of the right inequality of (1.3.2) where it must be such that all the constants and the function g (see Lamma 1.3.1) may be represented.

We first establish the relation between cost and length (1.3.1).

Any formula F over Φ can be built up from one which uses only one operator symbol (an elementary formula) by successively replacing variable symbols with new elementary formulas. If F does not contain one-argument operators, then whenever we increase the cost during the build-up (by adding an elementary formula with cost 1), we also increase the length. Specifically, the length increases by between $n_{min}-1$ and $n_{max}-1$ where n_{min} and n_{max} are respectively the smallest number larger than 1 and the greatest number of arguments of an operator of Φ. This results in the estimate

$$\frac{1}{n_{max}-1} \cdot L(F) \leq C(F) \leq \frac{c}{n_{min}-1} \cdot L(F) \qquad (1.3.3)$$

where c = 1. Suppose F contains one-argument operators. In other words, T(F) contains nodes with branching factor one. Let the maximal number of such nodes that occur one after another on any branch of T(F) be c*; then (1.3.3) still applies with c = c* + 1. (1.3.1) is obtained from (1.3.3) by noting that the minimal length or cost representation of any function (over the chosen basis Φ) can be achieved with a formula where $c^* \leq d^d$ (the number of functions $D \to D$).

The left inequality in (1.3.2) is established by a trivial counting argument (the maximal number of terminal nodes in a tree with branching factor $\leq n_{max}$ and depth d is n_{max}^d). The right side requires more effort (the following argument is due to R. W. Floyd). We first state the following obvious

1.3.1 Lemma.

Given a formula F such that $F = F_1(X_1, F_2(X_2))$ where F_2 is a proper subformula of F and $F_1 = F/F_2$, the following holds:

$$F \equiv F_3(F_1(X_1,C_0),\ F_1(X_1,C_1),\ldots,F_1(X_1,C_{d-1}),F_2(X_2))$$

where C_i for $D \leq i \leq d-1$ is any formula representing the constants $0,\ldots,d-1$ (or, as we have remarked previously, the one-argument function with constant value), and F_3 is any formula representing the function $g(z_0,\ldots,z_{d-1},z_{d-} =$ "z_0 if $z_d = 0$; z_1 if $z_d = 1,\ldots,$ z_{d-1} if $z_d = d-1$.

Let F be an arbitrary formula over Φ, and let G be a proper subformula of F. We already know that F = H(X,G). The claim is made that if $L(F) > 1$, G can be chosen in such a way that

$$L(H)-1,L(G) \leq \frac{n_{max}}{n_{max}+1} \cdot L(F) \qquad (1.3.4)$$

where n_{max} is as defined previously. (Remark: L(H)-1 is the number of occurrences of the variables of S*(G) in H.)

To find G use the following procedure: Start with F and proceed to subformulas of F. Assume you are considering the subformula K. Then two cases can arise. Either among $K_{.j}$ for $1 \leq j \leq k$ where k is the number of arguments of the outermost operator of K there is one, j', such that $L(K_{.j'}) \geq \alpha \cdot L(F)$ ($0 < \alpha < 1$ will be determined later with the purpose of obtaining the lowest possible estimate of L(H)-1 and L(G)), or not. In the first case, proceed to $K_{.j'}$ and containue. Otherwise, set $G = K_{.j''}$ where j'' is such that $L(K_{.j''}) = \max_{1\leq j\leq k} (L(K_{.j}))$ and terminate. Before the procedure terminates, $L(K) \geq \alpha \cdot L(F)$. Thus $\frac{\alpha}{n_{max}} \leq L(G) < \alpha \cdot L(F)$. This also means $(1-\alpha) \cdot L(F) \leq L(H)-1 < (1-\frac{\alpha}{n_{max}})$ $\cdot L(F)$ (because $L(G) + L(H)-1 = L(F)$). The lowest bound for L(G) and L(H) is obtained by setting $\alpha = 1 - \frac{\alpha}{n_{max}}$; hence (1.3.4)

Now apply Lemma 1.3.1 with G replacing F_2 and H replacing F_1. F_3 is of depth c, depending on Φ. If the outlined procedure is applied recursively to $H(X,C_i)$ for $0 \leq i \leq d-1$ and to G, we obtain in (1.3.2) $c_3 = \frac{c}{\log_2 b}$ where $b = \frac{n_{max}+1}{n_{max}}$. †

Note that unlike the cost-length relationship, the minimal value of depth may not be achieved by the same formula as the minimal value for length.

Apart from the relationship between the various measures, depth and cost will not be treated further. Even though in what follows (in this chapter) many things hold mutatis mutandis for depth and cost, most of the specific discussion and the examples shall be confined to length.

1.4 Problems Related to the Length Measure

In this section we will mention several questions that have been asked about the complexity of finite functions, their status as of this writing, and how they relate to the work to be described here.

---

† A more precise expression is obtained if the right side of (1.3.2) is replaced by $\frac{c}{\log_2 b} \cdot \log_2(L(f))+\ell$ for some constant ℓ. Namely, if we start out with a formula F and decompose it according to (1.3.4) and Lemma 1.3.1, then the length of $H(X, C_i)$ and G is bounded by $\frac{1}{b}L(F)+k$ where k is the length of C_i. After n applications of Lemma 1.3.1, the lengths of the relevant formulas are bounded by $\frac{1}{b^n} L(F) + k\frac{1}{b^{n-1}} +\ldots+ k\frac{1}{b}+k \approx \frac{1}{b^n} (L(F)) + \frac{k}{1-\frac{1}{b}}$; hence the figure above.

The Problem of Aggregate Length

Let Φ be a complete basis (for a certain domain D). The statement of the problem is: What is the largest number $L(n,\Phi)$ such that there exists a function $f: D^n \to D$ and $L(f,\Phi) = L(n,\Phi)$?

It has been studied by several authors, and is now effectively disposed of. Riordan and Shannon [Ri42] first derived a lower bound for $L(n,\Pi)$. Actually they studied series-parallel contact networks, but the two models are equivalent. The first upper bound (for the same model) was obtained by Shannon [Sh49]. Krichevskii [Kr59] derived a lower bound for $L(n,\Phi)$ for arbitrary domains and bases, while Lupanov [Lu59] obtained the best upper bound for the general case. The result is

$$O(L(n,\Phi)) = \frac{d^n}{\log_\alpha n} \tag{1.4.1}$$

where $O(f(n)) = g(n)$ means that $\lim_{n \to \infty} \frac{f(n)}{g(n)}$ is finite and nonzero. There are two remarks that are in order here. The first is that

> Formulas represent finite functions efficiently; i.e., the total number of formulas (over a given basis Φ) of length up to $L(n,\Phi)$ closely matches the number of functions of n variables. (1.4.2)

The second is

> The fraction of functions $D^n \to D$ that can be represented by formulas of length up to $L(n,\Phi)\cdot(1-\varepsilon)$ for an arbitrary $0 < \varepsilon \leq 1$ approaches zero as $n \to \infty$. (1.4.3)

The interested reader may obtain more information in the literature cited above.

Obviously, we could define functions $C(n,\Phi)$ and $D(n,\Phi)$ analogous to $L(n,\Phi)$ in terms of the cost and depth measures. In general, such functions (aggregate complexity functions) can be defined in connection with any model for the representation of functions $D^n \rightarrow D$ and any measure on this model (an obvious variation of $L(n,\Phi)$ is to remove the condition of completeness on Φ). It should be noted that the asymtotic behavior of aggregate complexity functions remains an active area of research. For references on the subject, see Lupanov [Lu70].

The Minimization Problem

Investigation of the complexity of finite functions started on representations of Boolean functions by logical circuits. In fact, formulas can be thought of as circuits with fan-out one. Thus, the first problems studied were those a logic designer is likely to ask: Given a finite function, what is the minimal circuit (formula) that represents it (i.e., find the complexity, and do so "effectively").

Unfortunately, no statisfactory solution to the minimization problem exists (for any measure). This does not mean that it is impossible to obtain a minimal formula for a given function $f: D^n \rightarrow D$; rather that existing algorithms are impractical. Thus, it is always possible to order formulas according to length, and then search all formulas up to length $L(n,\Phi)$ for the first formula that represents f; but since there are d^{d^n} functions of n arguments this approach is absurb.

At the present, all existing algorithms for the minimization of functional representations employ some sort of an exhaustive search (e.g., the Quine algorithm for the minimization of disjunctive form representations of Boolean

functions). In fact, there is reason to believe that a more efficient method does not exist, i.e.,

1.4.1 Conjecture

Any generally applicable exact minimization procedure is comparable (in terms of computational complexity) to an exhaustive search among formulas.

It is useful to consider a specific machine model. Let us consider implementations of such a procedure as a deterministic one-tape Turing machine M_{Φ} (see, e.g., Arbib [Ar69]) that receives as its input the d^n-tuple defining an arbitrary function f: $D^n \rightarrow D$, and whose output is the minimal formula F (over Φ) for f. Conjecture 1.4.1 gives us that the computation time of M_{Φ} may attain an exponential (in the length of the input). Let us venture a more restrictive and precise version of Conjecture 1.4.1:

1.4.2 Conjecture

Let M_{Φ} be as described, m is the length of its input, and let $\eta(m)$ be a function such that $\frac{\eta(m)}{c^m} \rightarrow 0$ as $m \rightarrow \infty$ for an arbitrary constant $c > 1$. Then the proportion of inputs of length m at which the running time of M exceeds $\eta(m)$ approaches 1 as $m \rightarrow \infty$.

Actually, the specific machine model on which the procedure of Conjecture 1.4.1 above is implemented is not particularly important. It can be easily shown (see, e.g., Arbib [Ar69], Chapter 4) that different deterministic machine models (this applies to the most widely used models, e.g., one-tape and multi-tape Turing machines) can simulate each other in such a way that the running

time of one is related to the running time of another at most by a polynomial. In this way, whenever the running time is exponential (in the length of the input) in one case, it must be so also in others.

It seems that Conjecture 1.4.1 was first expressed by Yablonskii [Ya59]. A very interesting result connected with this subject was recently obtained by Cook [Co71]. He obtained strong evidence that a simpler problem requires nonpolynomial time. The problem is that of recognizing whether a certain disjunctive normal form (for a Boolean function) represents the constant 1. Cook showed that if this problem could be solved in polynomial time (by a deterministic one-tape Turing machine), then a number of other problems that are regarded as very difficult (e.g., given the graphs G_1 and G_2, determine whether G_1 is isomorphic to a subgraph of G_2; the recognition of primes; etc.), would also be rapidly computable. Note that a fast minimization machine would give us also a fast constant recognizer; hence, Cook's results supports Conjecture 1.4.1.

The Classification Problem

In view of the difficulty of finding an exact nontrivial solution to the minimization problem (i.e., one that does not employ exhaustive search), present research is directed at establishing bounds for the length of functions. We consider sequences of functions $f_1,\ldots$ of $1,\ldots$ arguments and study the growth rate of the length of f_i. Thus, we can talk of classes of linear (length) sequences, quadratic (length) sequences, etc. Also of nonpolynomial (length) sequences. Unfortunately, if a sequence belongs to a nonlinear class, it is very difficult to estimate its length. We cannot even assign representatives to the polynomial classes of degree > 2, let alone the nonpolynomial

classes. In fact, at the present we have only a very limited store of examples of nonlinear sequences.

Consider the Boolean function $f_n^2 = \bigoplus_{i=1}^{n} x_i$. Subbotovskaya [Su61] gave a striking proof that $0(L(f_n^2, \Pi)) \geq n^{3/2}$. It was known already to Shannon (see [Sh49], or [Ya54]) that $0(L(f_n^2,\Pi)) \leq n^2$ (the length of this sequence, of course, grows linearly if $\oplus$ is used). Unfortunately, it seems that the technique of [Su61] cannot be generalized to $d > 2$. Subbotovskaya's result has recently been improved by Khrapchenko [Kh71]. He succeeded in showing that $0(L(f_n^2, \Pi)) \geq n^2$. Since this result employs a very interesting technique, and since it has not yet been translated into English, it is reproduced in Appendix B.

Neciporuk [Ne66] discovered a sequence of Boolean functions f_n such that $0(L(f_n,\Phi)) = \frac{n^2}{\log_2 n}$ for an arbitrary basis Φ. It is true that the functions involved in the Neciporuk sequence are rather "artificial" in that, while defined in a straightforward way, they have no special significance; however, lately Harper and Savage [Ha71] have succeeded in applying the Neciporuk technique to a practical combinatorial problem (The Marriage Problem).

Neciporuk's construction is based on the following lemma: Let f be a Boolean function of n arguments. Consider a subset X of the arguments of f and the set of restrictions of f to X obtained by setting the arguments outside of X to constants. Let the number of such restrictions be r. If F is any formula over a finite basis Φ for f, then the number of occurrences of variables representing the arguments in X is $\geq c \cdot \log_2 r$ where c depends on the basis Φ (for the proof of this see [Ne66] or [Ha71]).

The Neciporuk function f_n of n arguments is then obtained as follows: The n arguments are arranged in a rectangular array with dimensions as shown in Fig. 1.2. Each argument x_{ij} is associated with a 0-1 valued m-tuple $\underline{a}_{ij}$

such that (1) not all components are 0, and (2) if $(i,j) \neq (k,\ell)$ then $a_{ij} \neq a_{k\ell}$. Then we define

$$f_n = \bigoplus_{\text{all } i,j} x_{ij} \quad \bigoplus_{i \neq j} K(\underline{a}_{ij}, k)$$

where $K(\underline{a}_{ij},k)$ denotes the conjunction of those arguments x_{kt} whose second subscript (t) corresponds to nonzero components of $\underline{a}_{ij}$.

It can be verified that the number of restrictions of f_n to the variables of an arbitrary row (except, perhaps, the last which may be imcomplete) obtained by replacing the variables of the other rows with constants is 2^{n-m}. This follows from the fact that any Boolean function can be uniquely represented by a Boolean polynomial (see Lemma 4.5). Then, by the lemma above, the number of occurrences of variables of any row (except, perhaps the last) in any formula for f_n is $\geq c \cdot (n-m)$; hence, the length of f_n over $\Phi \approx c \cdot \frac{n}{m} (n-m)$. In other words, $O(L(\varphi_n,\Phi)) = \frac{n^2}{\log_2 n}$ for an arbitrary basis Φ.

Neciporuk's construction may be viewed as a solution to a special case of the following problem (the problem of exhibiting a function of arbitrary length): Given a basis Φ and a number $k \leq L(n,\Phi)$, exhibit a function $f: D^n \to D$ of length $\geq k$ over Φ. In Neciporuk's case $O(k) = \frac{n^2}{\log_2 n}$.

Since so few examples of functions that are known to be of large length exist (in spite of (1.4.3)), the reader has no doubt already gained the impression that this problem too is very difficult. However, we again have the trivial solution that consists in examining formulas in n variables in the order of their length, recording the functions they represent, and choosing the first previously unencountered function represented by a formula of length $\geq k$. In fact, it is reasonable to state an analog of Conjecture 1.4.1:

1.4.3 Conjecture

The problem of exhibiting a function of arbitrary length is comparable (in terms of computational complexity) to an exhaustive search among formulas.

We again make this conjecture more precise on the example of deterministic one-tape Turing machines.

1.3.4 Conjecture

Φ is an arbitrary basis. N_Φ is a deterministic one-tape Turing machine with input (n,k) where n is arbitrary and $k \leq L(n,\Phi)$, and whose output is the d^n-tuple describing a function f of n arguments such that $L(f,\Phi) \geq k$. Then there exists a constant $c > 1$ such that if $k \geq \epsilon \cdot L(n,\Phi)$ for any $0 < \epsilon \leq 1$, the running time of N_Φ on input (n,k) exceeds c^{c^n} when $n \geq n(\epsilon)$.

We can sum up the discussion of the classification problem as follows. The problem is far from understood. At the present no sequence of functions is known whose length grows faster than n^2. Isolated examples of sequences with growth rate $\leq n^2$ are known, and present research is directed at inventing more general techniques that can be used for estimating the complexity of whole classes of sequences. Also techniques have to be devised for $d > 2$. The importance of this will be discussed below in Section 1.5.

1.5 Specker's Theorem

The first general technique for proving the nonlinearity of a large class of sequences (of Boolean functions) was discovered by Specker [Ho68]. Let the basis $\Pi \cup \{x \oplus y\}$ be denoted by Σ. Then

Theorem (Specker).

If f is a Boolean function of n arguments, if $L(f,\Sigma) \leq c \cdot n$ for some constant c, then for any integer m, if $n \geq \eta_S(m,c)$, a subset $X = \{x_1,\ldots,x_m\}$ of the arguments of f can be found such that (1)

$$f_0^X = c_0 \oplus c_1 \cdot \prod_{i=1}^{m} (1 \oplus x_i) \oplus c_2 \bigoplus_{i=1}^{m} x_i \qquad (1.5.1)$$

where c_0, c_1, c_2 are Boolean constants and $\eta_S(m,c)$ is a certain number-theoretic function.† Furthermore, (2) if the basis is Π (the other assumptions remaining unchanged), then $c_2 = 0$.

This theorem has been used by Hodes and Specker to show that the predicate

$$\sum_{i=1}^{n} x_i \equiv 0 \bmod k \qquad (1.5.2)$$

for $k > 2$ and $x_i \in \{0,1\}$ is of nonlinear length over Σ.

Using the second statement of the theorem, they are also able to give an alternative proof of the nonlinearily of the length of $\bigoplus_{i=1}^{n} x_i$ over Π.

Another result obtained with Specker's Theorem is the fact that some geometrical predicates (in particular, connectivity) discussed by Minsky and Papert [Mi69] are of nonlinear length over Σ (see Hodes [Ho70]).

In Chapter Two we will formulate and prove a generalization of Specker's Theorem (Theorem 2.2.2) to include the case $d > 2$ and multi-argument operators in Φ. Our proof reveals the nature of both results more clearly.

† η_S Will be discussed in Chapter Two.

They belong to a class of combinatorial results reminiscent of Ramsey's Theorem (see Ryser [Ry63]). In fact, an earlier version of our proof of Theorem 2.2.2 used Ramsey's Theorem. Besides this, Theorem 2.2.2 enables us to derive the nonlinearity of new functions (sequences of functions) such as counting mod p where p is a prime, d possibly equals p, but there are restrictions on the basis, etc. An example of an improvement over existing results is the connectivity predicate. Hodes [Ho70] proves that it is nonlinear if d = 2. However, in Automata Theory, for example, the result that a certain language can be computed in nonlinear time if k states are used in the finite control would be considered weak. Rather we search for proofs that work for arbitrary finite controls. The Generalized Specker Theorem (Theorem 2.2.2) gives us a tool for proving the nonlinearity of the length of the connectivity predicate regardless of the domain D and basis Φ. We can apply it to connectivity by "reducing" connectivity (for the meaning of "reduction" see [Mi69] or 3.2) to certain symmetric functions.

We should note that the generalization of Specker's Theorem that we prove is the obvious one to attempt; but, as the reader will see, the proof turns out to be less straightforward. As an indication, consider (1.5.2). It does not generalize directly to $d > 2$ since, e.g., the function $\{0,1\}^n \to \{0,1\}$ defined by $\sum_{i=1}^{n} x_i \equiv 0 \bmod 6$ can be represented in linear length with $d = 3$. This is because

$$\left[\sum_{i=1}^{n} x_i \equiv 0 \bmod 6\right] = \left[\sum_{i=1}^{n} x_i \equiv 0 \bmod 3\right] \wedge \left[\sum_{i=1}^{n} x_i \equiv 0 \bmod 2\right]$$

Hodes and Specker do not derive any bounds for the lengths of the functions investigated by them. This question is asked (and to an extent answered) in 3.3.

1.6 Cyclic Perceptrons

Cyclic Perceptrons will be treated in Chapter Four. They are an application of ideas of Minsky and Papert to the representation of functions by combinations of finite operators. In particular, one of the concerns in [Mi69] is to formalize the intuitive idea that the connectivity predicate, being "global" in nature, cannot be computed (or represented) by a "simple" combination of "local" predicates.

The perceptron is the predicate

$$\sum_{i \in I} a_i \cdot \varphi_i \geq 0$$

where I is an indexing set, $a_i \in Q$, the rationals, $\varphi_i \in \Phi$, a set of Boolean functions (whose value is interpreted as being either the rational 0 or 1. The cyclic perceptron is defined as

$$\sum_{i \in I} a_i \cdot \varphi_i \in Y$$

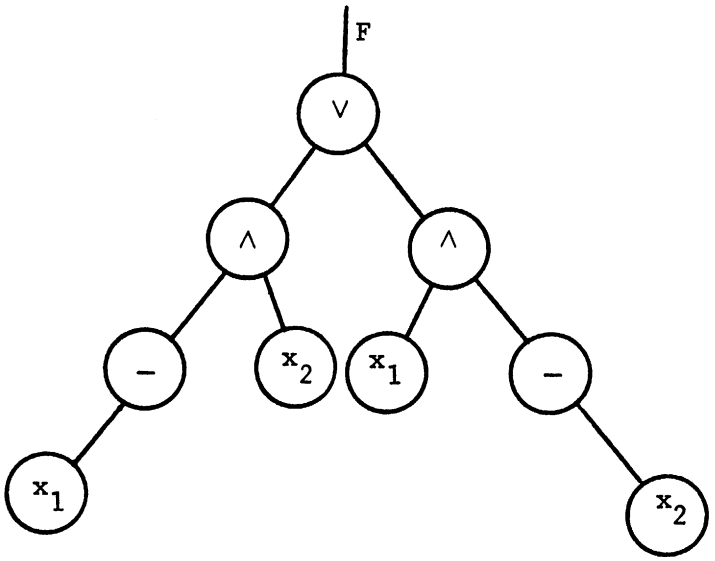
where $a_i \in F$, a finite field, $Y \leq F$, and other symbols have the same interpretation as before. Thus, both represent a certain Boolean function.

Minsky and Papert introduce the concept of the order of a perceptron (the maximal number of arguments on which φ_i depends where i ranges over I). They define then the order of a predicate as the minimal order of a perceptron that represents the predicate. They formalize "local" by defining

an infinite predicate sequence to be local if and only if every member is representable by a perceptron of order $\leq r$, for some finite r. They are then able to show that connectivity is nonlocal.

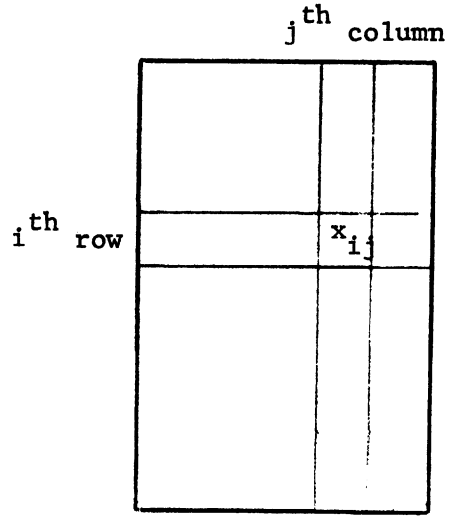
The concept of order can also be applied to cyclic perceptrons. Chapter Four will contain results on the order of the various predicates introduced in [Mi69]. In particular, connectivity is shown to be nonlocal. This will be an extension (to finite fields of arbitrary characteristic) of the results described in [Vi70].

Chapter Five describes a model of computation (Pattern Counting Machines) that again performs a "local" computation followed by a "global" computation. In this case the "local" computation is even more constrained than in the case of perceptrons. The result is that no matter how cleverly we utilize the "local" information in the subsequent "global" phase, the connectivity predicated cannot be computed.



T(F) for the formula F in Example 1.2.2

Fig. 1.1



number of columns: $m = \lceil \log_2 n \rceil + 1$

number of rows: $\lceil n/m \rceil$

The array of arguments used in the definition of the Neciporuk function f_n

Fig. 1.2

CHAPTER TWO

A GENERALIZATION OF A THEOREM OF SPECKER

2.1 e-Complexes

Throughout this section, all formulas are D-formulas for some fixed (but arbitrary) domain D, and all operators are functions $D^r \to D$.

Given the formulas $F_1,\ldots,F_r$, we shall call the formula $F = \varphi(F_1,\ldots,F_r)$ where φ is an arbitrary operator a *parallel combination* (PC) of $F_1,\ldots,F_r$. φ is called the *decoding* operator of F.

Let F(X,z) be a formula where the distinguished variable z appears only once, and let G be an arbitrary formula. Then F(X,G) shall be called a *series combination* (SC) *of* F *and* G *through* z.

2.1.1 Definition

We give an inductive definition of an *elongated* n-*component* (e_n-component) for $n \geq 0$.

(1) Let φ_{in} be an arbitrary unary operator and z an arbitrary variable symbol. Then $\varphi_{in}(z)$ is an e_0-component. z is the *input variable* while φ_{in} is the *input operator*.

(2) Let φ be an arbitrary binary operator, G an arbitrary e_{n-1}-component, and $x \notin S(G)$. Then $F = \varphi(x,G)$ (or $\varphi(G,x)$) is an e_n-component. The input variable and input operator of G are also the input variable and input operator of F. x is a *lateral variable* of F. Any lateral variable of G is also a lateral variable of F. φ will be called an *internal* operator of F.

An example of an e_n-component is given in Fig. 2.1. Let F be an arbitrary e_n-component, and let $\underline{x}$ be the sequence of lateral variables arranged in the order they are connected to the branch of T(F) extending to the input variable. Then $\underline{x}$ is the lateral sequence of F. If F is an e_0-component, then the lateral sequence of F is λ (the empty sequence). For example, the lateral sequence of the e-component in Fig. 2.1 is $x_1,\ldots,x_n$.

An e_n-component with all internal operators equal is a homogeneous e_n-component.

2.1.2 Definition

A formula F is an e_n^r-complex if (1) F is a PC of the e_n-components $F_1,\ldots,F_r$, and (2) the lateral sequence of F_i for $2 \leq i \leq r$ is either equal to the lateral sequence of F_1, or the reverse of it.

$F_1,\ldots,F_r$ are the components of F. If the variables of F_1 are numbered as in Fig. 2.1, the second condition of Definition 2.1.2 means that any component $F_1,\ldots,F_r$ either appears as in Fig. 2.1, or as in Fig. 2.2. The components of the former kind will be known as standard components, while those of the latter kind will be called the reverse components. The lateral sequence of F_1 will also be called the lateral sequence of F.

Both in the case of e_n-components and e_n^r-complexes, one or both indices will occasionally be omitted if the particular property they refer to is irrelevant to the argument at hand.

An e-complex composed of homogeneous e-components is a homogeneous e-complex.

One might wonder what the purpose of introducing e-complexes is since for appropriate r and m every function of n variables can be represented by an e_m^r-complex. Thus, it would seem that this class of formulas is trivial. However, we will be concerned with e_n^r-complexes where r remains fixed as n grows without bounds, and this will allow us to obtain interesting results.

We introduce some notation. Let F be an e_n-component with lateral sequence $x_{i(1)}, x_{i(2)}, \ldots, x_{i(n)}$. $a \in D$ is an arbitrary constant. φ_j denotes the internal operator corresponding to $x_{i(j)}$. Also set $\varphi_{n+1} = \varphi_{in}$. Then

$$\varphi^a_{(j,k)} = \begin{cases} \varphi_j(a,\varphi_{j+1}(a,\ldots,\varphi_{k-1}(a,\varphi_k)\ldots)) \text{ if } 1 \leq j < k \leq n+1 \\ \varphi_j \text{ if } 1 \leq j = k \leq n+1 \\ \text{undefined otherwise} \end{cases}$$

Note that $\varphi^a_{(i,j)}$ is a unary operator if $j = n+1$, otherwise it is a binary operator (if it is defined at all). Usually we will suppress the superscript a because it will be clear what constant is referred to.

We now state the simple

2.1.3 Proposition

Let F be an e_n-component with lateral sequence $\underline{x}$ and input variable z. $\underline{y}$ is an arbitrary subsequence of $\underline{x}$ of length $m \geq 0$ and $a \in D$ is an arbitrary constant. If we denote the set consisting of z and the elements of $\underline{y}$ by Y, then F_a^Y is equivalent to an e_m-component G.

Proof

Let $\underline{x} = (x_1, x_2, \ldots, x_n)$ and $\underline{y} = (x_{i(1)}, x_{i(2)}, \ldots, x_{i(m)}) \subseteq \underline{x}$. Set $i(0) = 0$ and $i(m+1) = n+1$. Then G has the operators $\psi_j = \varphi_{(i(j-1)+1,\ i(j))}$ for $i \leq j \leq m+1$ (ψ_{m+1} is the input operator of G). □

2.1.4 Remark

Obviously, Proposition 2.1.3 holds for e-complexes as well; one merely has to perform the above construction for each component.

Proposition 2.1.3 will be frequently invoked. Namely, we will take an e-complex F, select a subsequence $\underline{y} \subseteq \underline{x}$, the lateral sequence of F, and obtain G as above. In this case, G is called the result of an a-merger with basis y on F.

We introduce another restricted class of formulas.

2.1.5 Definition

A series parallel combination of e-components (SPCeC) is obtained according to the following rules:

(1) An e-component is an SPCeC.

(2) Let F and G be an e-component and an arbitrary SPCeC respectively. Then the SC of F and G through the input variable of F is an SPCeC.

(3) If $F_1, \ldots, F_r$ are SPCeC's, then a PC of $F_1, \ldots, F_r$ is an SPCeC.

(4) An SPCeC is only an object satisfying (1), (2), or (3).

Given an arbitrary SPCeC F, we describe its set of components. If F consists of the single e-component G, then G is the only component of F. If F is the SC of an e-component G and another SPCeC H, then the set of components of F consists of G and the set of components of H. If F is a PC of $F_1,\ldots,F_r$, then the set of components of Fconsists of the sets of components of F_i for $1 \leq i \leq r$. Among the components of F, those whose input variable corresponds to a terminal node of T(F) will be called terminal components while the others will be called internal components. An example of an SPCeC is given in Fig. 2.3. This particular SPCeC has four terminal components and two internal components.

2.1.6 Proposition

An SPCeC is equivalent to a PC of r e-components where $r = d \cdot I+J$ and I and J respectively are the number of internal and the number of terminal component of F.

Proof F can be converted into a PC of e-components by using Lemma 1.3.1. The estimate of the number of e-components in the PC is also obtained from there. □

Remark It is a simple matter to verify that if F of Proposition 2.1.6 has k components, then $I \leq k-1$; and thus $r \leq d \cdot (k-1)+1$.

2.1.7 Proposition

F is a SPCeC with k components $F_1,\ldots,F_k$. F_i for $1 \leq i \leq k$ is an e_n-component for $n \geq 0$, and, furthermore, the sets of lateral variables of F_i and F_j are equal for $1 \leq i, j \leq k$. Let X be the set of lateral variables of F_i and let Z be the set of input variables of F. Then for any $m \geq 0$ and

$a \in D$ if $n \geq \eta_1(m,k)$ where $\eta_1(m,k)$ is a certain function (to be defined), there exists a subset $Y \subseteq X$ with $|Y| = m$ such that F_a^{YUZ} is equivalent to an e_m^r-complex G with Y as the set of lateral variables. Furthermore, $r \leq d\cdot(k-1)+1$.

Proof If $m = 0$, we can immediately apply Proposition 2.1.6 and obtain an e_0^r-complex where r is as described in the statement of the proposition; thus $\eta_1(0,k) = 0$. We assume, therefore, that $m > 0$.

We recall the following familiar result:

Let $i(1), i(2),\ldots,i((p-1)^2+1)$ be a sequence of distinct integers. Then we can extract a subsequence of length p that is either increasing or decreasing (for the proof see Berge [Be71] p. 16).

Without loss of generality, we can assume that the lateral sequence of F_1 is $x_1,\ldots,x_n$. Then the lateral sequence of F_2 is $x_{i(1)}, x_{i(2)},\ldots,x_{i(n)}$. The sequence $i(1), i(2),\ldots, i(n)$ consists of distinct integers; therefore, if $n \geq (n_1-1)^2+1$, we can apply the above result and find a subset $X_1 \subseteq X$ of n_1 variables such that after performing an a-merger with basis X_1 on all components of F, the lateral sequences of the descendants of F_1 and F_2 are either the same or opposite. We can continue in this way, processing one after another all components. We end up with an SPCeC with components $G_1,\ldots,G_k$ such that the lateral sequence of G_i for $2 \leq i \leq k$ is either equal to that of G_1 or the reverse of it. To obtain G, we apply Proposition 2.1.6. In order that $|Y| = m$, we must have

$$n \geq \eta_1(m,k) = (m-1)^{2^{k-1}}$$

for $m \geq 1$. The estimate for r is obtained from the Remark following Proposition 2.1.6. □

Another equivalence that will be used later is given by

2.1.8 Lemma

E is an e_n^r-complex, X is the set of its lateral variables, and Z is the set of its input variables. Then for any $m \geq 0$ and $a \in D$, if $n \geq \eta_2(m,r)$, there exists a subset $Y \subseteq X$ with $|Y| = m$ such that $E_a^{Y \cup Z}$ is equivalent to a homogeneous e_m^r-complex F.

Proof If $m = 0$, we simply use Proposition 2.1.3, and the result is a homogeneous e_0-complex (obviously, any e_0-complex is homogeneous). Then $\eta_2(0,r) = 0$. Thus, from now on we assume that $m \geq 1$.

The proof will be given for the special case when E has two components: a standard component E_1 and a reverse component E_2. It will then only be indicated how to generalize the proof.

A procedure (The Homogenizing Procedure--HP) will be described that will transform an e_p^2-complex G consisting of a standard component G_1 and a reverse component G_2 with $p \geq \eta_3(q)$ (for a function η_3 that will be defined later) and with the properties: (1) There exist (possibly empty) subsets R_1 and $R_2 \subseteq D$ such that $\omega_i(a,y) \upharpoonright R_1 = id_{R_1}$ (identity on R_1) and $\psi_i(a,y) \upharpoonright R_2 = id_{R_2}$ for $1 \leq i \leq p$ where ω_i and ψ_i is an operator of G_1 and G_2 respectively and the first argument corresponds to the lateral variable, and (2)

$$\forall(1 \leq i, j \leq p)\ [\omega_i(x,y) \in R_1 \Rightarrow \omega_j(x,y) = \omega_i(x,y)] \qquad (2.1.1)$$

(i.e., the operators of G_1 are identical on the inverse image of R_1). Similarly for the operators of G_2 on R_2.

Remark Note that if R_1 and R_2 include the range of every operator, Property 2 translates into the identity of the operators. In particular, this holds if $R_1 = R_2 = D$.

Remark Note that an arbitrary e^2-complex satisfies Properties 1 and 2 with $R_1 = R_2 = \phi$.

The result of applying HP will be an e_q^2-complex H that will either be homogeneous, or will have Properties 1 and 2 with S_1 and S_2 replacing R_1 and R_2 respectively and $R_1 \subsetneq S_1$ or $R_2 \subsetneq S_2$. Due to the Remarks above and to the fact that D is finite, repeated application of HP on E finally yields F. The condition on n is

$$n \geq \eta_2(m,2) = \underbrace{\eta_3(\eta_3(\ldots\eta_3(m)\ldots))}_{2d \text{ times}} \tag{2.1.2}$$

This bound for n corresponds to the worst case when R_1 or R_2 increase by only 1 on each application of HP.

Before describing HP, note the useful fact that because of Property 1, Properties 1 and 2 are preserved under a-mergers.

Description of HP The lateral sequence of G is of length $(v+1)\cdot u-1$ for certain values of u and v that will be defined later.

Consider the sequence

$$(\varphi^a_{((k-1)\cdot u+1,k\cdot u)}, \psi^a_{((v-1)\cdot u+1,\ (v-k+1)\cdot u)}) \tag{2.1.3}$$

for $k = 1,\ldots,v$. Sequence (2.1.3) is illustrated in Fig. 2.4. The two vertical lines represent G_1 and G_2; the numbered horizontal outlets represent the lateral variables (with the corresponding number); the boxes indicate the

variables and operators that take part in the formation of any particular $\varphi_{(i,j)}$ and $\psi_{(i,j)}$; an 'x' beside a variable indicates that it is not set to the constant while 'a' indicates that it is set to a; the two checked boxes represent the first member of (2.1.3).

In the sequence (2.1.3), either (Case I) the ranges of $\varphi_{((k-1)\cdot u+1,k\cdot u)}$ and $\psi_{((v-k)\cdot u+1,(v-k+1)\cdot u)}$ for $1 \le k \le v$ are included in R_1 and R_2 respectively, or (Case II) not.

Case I. If v is large enough, we can find q identical elements in the sequence (2.1.3). Let the indices k corresponding to these elements be $k_1,\ldots,k_q$. Performing an a-merger with this set as basis, the desired e-complex H is obtained. Note that in this case we use Property 1 of G. Namely if φ^* is the first component of a pair in (2.1.3) whose range $\subseteq R_1$ and if φ is an arbitrary operator of G_1, then $\varphi(a,\varphi^*) = \varphi^*$ (similarly for the second components of the pairs in (2.1.3) and operators of G_2). Thus, the components of the identical pairs in (2.1.3) become the operators of H.

A bound for v is $q\cdot(d^{d^2})^2$ (d^{d^2} is the number of operators $D^2 \to D$).

Case II. Assume $\varphi_{((\ell-1)\cdot u+1,\ \ell\cdot u)}(b,c) \notin R_1$ for some b, c $\in$ D and $1 \le \ell \le v$ (the case when $\psi_{((v-\ell)\cdot u+1,\ (v-\ell+1)\cdot u)}(b,c) \notin R_2$ can be treated similarly). But then

$$\varphi_{(\ell\cdot u-j,\ \ell\cdot u)}(b,c) \notin R_1 \qquad (2.1.4)$$

for all $0 \le j \le u-1$ (as a consequence of Property 1). Provided that u is large enough, we can find an element e $\in$ D that appears w times in the sequence (2.1.4). Let the indices j corresponding to the appearances of e be $j(1),\ldots,j(w)$ (see Fig. 2.5). Obviously then (all the variables considered except $x_{\ell\cdot u}$ have been set to a),

$$\varphi_{(\ell\cdot u-j(t),\ \ell\cdot u-j(t-1)-1)}(a,e) = e$$

for $2 \leq t \leq w$ (the first argument of $\varphi_{(i,j)}$ corresponds to the lateral variable). Thus,

$$\varphi_{(\ell\cdot u-j(t),\ \ell\cdot u-j(t-1)-1)}(a,y)\restriction R_1 \cup \{e\} = id$$

At this point we consider separately two cases:

Case IIa $\ell = 1$ and $j(w) = u-1$. We perform an a-merger with the basis consisting of the variables with indices $u-j(t)-1$ for $1 \leq t \leq w-1$. As a result of this we obtain an e^2_{w-1}-complex G' such that Property 1 holds for G'_1 on $R_1 \cup \{e\}$ (G'_2 satisfies Property 1 on $R'_2 \subseteq R_2$; in any event $R_2-R'_2 = \phi$). Note that at this point Property 2 is still satisfied only on R_1 and R_2 by G'_1 and G'_2 respectively.

Case IIb. $\ell \neq 1$ or $j(w) < u-1$. We perform an a-merger with the basis consisting of the variables with indices $\ell\cdot u-j(t)-1$ for $1 \leq t \leq w$. As a result of this we obtain an e^2_w-complex G'' such that Property (almost) holds for G''_1 (the same remarks regarding G''_2 and Property 1 as well as G'''_1, G'''_2 and Property 2 apply as in Case IIa). The only exception may be φ''_1 (the operator of G''_1 that is closest to the decoding function). By definition, $\varphi''_1 = \varphi_{(1,\ell\cdot u-j(w)-1)}$, and there is no assurance that $\varphi''_1(a,e) = e$. We may rectify this situation by absorbing φ''_1 into the decoding function in the following way: Let $G''' = \theta(G'''_1, G'''_2)$. Now set $x_1 = a$ (after the a-merger the variables have been renumbered). Let $S(G'') - \{x_1\} = U$. Then we have $(G''')^U_a = G' = \theta(\varphi'''_1(a,G'_1), G'_2)$ where G'_1 equals G'_1 minus φ'_1 and G'_2 equals G'_2 with the input operator modified as follows: $\psi'_{in} = \psi''_w(a,\psi''_{in})$ (remember that G''_2 is a reverse components, and, hence, x_1 is attached to ψ''_w). Clearly, G' is an e^2_{w-1}-complex satisfying

Property 1 with $R_1 \cup \{e\}$ replacing R_1.

We can now resume considering Cases IIa and b together. To obtain H (with components H_1 and H_2) we must find among φ_i' for $1 \le i \le w-1$ q operators that are identical on the inverse image of $R_1 \cup \{e\}$ (i.e., (2.1.1) with $R_1 \cup \{e\}$ replacing R_1) and again perform an a-merger. We again emphasize that the operators of H_2 are identical only on the inverse image of R_2, and this property has not been violated by any of the transformations of the original e-complex G.

To obtain q operators that are identical on the inverse image of $R_1 \cup \{e\}$, it is sufficient that $w-1 \ge q \cdot d^{d^2}$; therefore, $u \ge d \cdot (q \cdot d^{d^2}+1)$ and

$$\eta_3(q) = q^2 \cdot d \cdot \delta^3 + q \cdot d \cdot (\delta^2+\delta)5d-1 \qquad (2.1.5)$$

where $\delta = d^{d^2}$. This is obtained from the values of u and v derived above. Recall that $\eta_3(q) = (v+1) \cdot u-1$, the length of G. η_2 for r = 2 can then be obtained from (2.1.5) and (2.1.2).

The proof for the general case is obtained by defining the Generalized H Procedure (GHP) with the corresponding function η_4. We consider instead of (2.1.3) the sequence

$$(\varphi^1_{(s,t)},\ldots,\varphi^{r'}_{(s,t)}, \psi^1_{(s',t')},\ldots,\psi^{r''}_{(s',t)})$$

where $s = (k-1) \cdot u+1$, $t = k \cdot u$, $s' = (v-k) \cdot u+1$, and $t' = (v-k+1) \cdot u$. $\varphi_i^1,\ldots,\varphi_i^{r'}$ denote the operators of the standard components of G while $\psi_i^1,\ldots,\psi_i^{r''}$ denote the operators of the reverse components ($r' + r'' = r$). Without detailed argument we state that in the general case $v = q \cdot \delta^r$ while u remains the same (u is determined by the requirements of Case II at which time only one

component is considered). From this we obtain analogously to (2.1.5)

$$\eta_4(q) = q^2 \cdot d \cdot \delta^{r+1} + q \cdot d \cdot (\delta^r + \delta) + d - 1$$

$\eta_2(m,r)$ can then be obtained from

$$\eta_2(m,r) = \eta_4(\underbrace{\eta_4(\ldots\eta_4(m)\ldots)}_{r \cdot d \text{ times}})$$

which is an analog of (2.1.2). □

2.1.9 Remark

As we have seen, η_2 in Lemma 2.1.8 depends on r, the number of components of F. However, we shall mostly be using e-complexes that contain many components that are identical except for the input operator (such e-complexes are obtained e.g., by the use of Proposition 2.1.7). It may be checked that in the application of GHP only one representative from each such group of components need be considered. This significantly reduces η_4. Similarly, in computing η_2 from (2.1.5), only $\ell \cdot d$ compositions are required where ℓ is the number of groups of similar components (corresponding to d compositions for each group of similar components).

2.1.10 Remark

The operators of a homogeneous e_m^r-complex F obtained as a result of applying Lemma 2.1.8 possess an added property that will be used later: Let R be the range of $\varphi(x,y)$ an internal operator of R (x corresponds to a lateral variable); then $\varphi^i(a,y)$ $R = id_R$ (the identity on R). This fact follows from the definition of HP (GHP). This particular property of the operators of the components of F will be called the I_R^a-property relative to y. In what

follows, we will always suppress "relative to y" since there is no danger of ambiguity. We will simiarly suppress the subscript R unless we will be interested in a specific range. We will abbreviate "F is an e-component (complex) whose operators possess the I^a-property" to "F is an e-component (complex) with the I^a-property".

A familiar and convenient way of representing a binary operator $\varphi(x,y)$ is by a labeled directed graph. The graph of φ, denoted by $\Gamma(\varphi)$, is defined as follows: The nodes of $\Gamma(\varphi)$ are labeled with elements of D. A directed arc labeled with a $\in$ D exists from b to c if and only if $\varphi(a,b) = c$.

If D = {1, 2, 3, 4} and R = {2, 3, 4} an example of a graph $\Gamma(\varphi)$ for an operator φ with the I_R^2-property is shown in Fig. 2.6.

Given an arbitrary e_n-component F, the output of the operator φ_k is

$$\varphi_k(x_k,\varphi_{k+1}(x_{k+1},\ldots,\varphi_n(x_n,\varphi_{in}(y))\ldots));$$

in the case of a homogeneous e_n-component with internal operator φ this will be abbreviated to

$$\varphi(x_k x_{k+1}\cdots x_n,\ \varphi_{in}(y)).$$

2.2 The Generalized Specker's Theorem

We first give the following

2.2.1 Definition

Let Φ be an arbitrary basis and a $\in$ D any constant. Let F be a formula over Φ with $S(F) = X \cup Y \cup Z$ such that $|X| \leq n_{max}-1$ (the maximal number of arguments of an operator of Φ), Y is disjoint from X, but otherwise arbitrary,

and $Z = \{z\}$ is a singleton (disjoint from X and Y) such that z occurs only once in F. The set of functions $f(X,z)$ represented by all possible such formulas F with the elements of Y replaced by the constant operator a will be denoted by Φ^a.

In particular, every operator of Φ with all but $k \geq 1$ arguments (k is arbitrary) replaced by a is in Φ^a. Note that if $\varphi(X,z) \in \Phi^a$, then φ may qualify for Φ^a by virtue of a number of different representations. If z (or any variable of X) in any one of them corresponds to a variable that occurs only once, it is called a distinguished argument). The other arguments are called free arguments. Thus we may easily find a basis Φ and an operator φ such that all arguments are at the same time distinguished and free.

We now define a restricted class of e-components and e-complexes: Φ is an arbitrary basis and $a \in D$ is any constant. Let F be an arbitrary e_0-component; then F is an e_0-component over Φ^a. Let $\varphi(x,z) \in \Phi^a$ be a binary operator, z a distinguished argument (hence x is free), and G an e_{n-1}-component over Φ^a; then $\varphi(x,G)$ is an e_n-component over Φ^a. An e-complex over Φ^a is an e-complex such that all its components are e-components over Φ^a.

The main result of this chapter is

2.2.2 Theorem

Let there be given the function $f: D^n \to D$ such that $L(f,\Phi) \leq c.n$ for some constant c and basis Φ. Then for any $m \geq 1$ and $a \in D$ if $n \geq \eta_5(c,m)$, there exists a subset Y of the arguments of f such that $|Y| = m$ and f_a^Y is either a constant or is represented by F, a homogeneous e_m^r-complex over Φ^a with the I^a-property, Y as the set of lateral variables, constant input operators, and $r \leq d\cdot(2c-1)+1$.

Proof

If f has m fictitious arguments, then let Y be the set of these arguments, and f_a^Y is a constant. From now on we assume that f has $\leq m-1$ fictitious arguments.

The statement of the theorem gives us that there exists a formula E over Φ such that $L(E) \leq c \cdot n$. Therefore, there are $\geq 1/2 \cdot n$ variable symbols representing the arguments of f which either do not appear in E or appear $\leq 2 \cdot c$ times. In other words, there are $\geq 1/2n-m+1$ variable symbols that actually appear in E and such that the number of occurrences of each is $\leq 2 \cdot c$. Denote the set of these variables by X_1.

If $n \geq 2 \cdot \eta_8(n_2, 2c)+m-1$, we can apply Lemma A.9 and obtain a subset $X_2 \subseteq X_1$ with $|X_2| = n_2$ and such that $E_a^{X_2}$ is equivalent to E_2, and SPCeC over Φ^a with at most 2c components and such that the set of lateral variables of every component is X_2.

If $n_2 \geq \eta_1(n_3, 2c)$ we can apply Proposition 2.1.7 and obtain a subset $X_3 \leq X_2$ with $|X_3| = n_3$ and such that $E_a^{X_3}$ is equivalent to E_3, an $e_{n_3}^r$-complex over Φ^a with $r \leq d \cdot (2c-1)+1)$. The estimate for r is obtained at this point.

If $n_3 \geq \eta_2(m, 2c)$ we can apply Lemma 2.1.8 and Remark 2.1.9 and obtain F, the desired homogeneous e_m^r-complex over Φ^a with the I^a-property. The I^a property is a consequence of Lemma 2.1.8.

Discussion of η_5. The present proof yields

$$\eta_5(m,c) = 2\cdot\eta_8(\eta_1(\eta_2(m,2c),2c),2c)+m-1$$

The exact representation of η_5 is extremely complex, and in what follows we shall use only a very rough approximation. In Appendix A it is seen that $\eta_8(t,k)$ (as a function of k) grows faster than iexp(b,2k) for any constant b. The functions η_1 and η_2 contribute only insignificantly to this, and thus we state:

$$\eta_5(m,c) \geq \text{iexp}(b,4c) \text{ for } c \geq c(b) \text{ and an arbitrary constant } b \qquad (2.2.1)$$

(Later we shall se that the size of η_5 prevents us from obtaining any interesting bounds for the functions investigated with Theorem 2.2.2. The size of η_8 which contributes most to η_5 results from the technique used in Lemma A.3 to obtain a nesting sequence for a given formula F. It is not known whether this technique can be improved. Our guess is that it cannot be.) □

2.3 On Specker's Theorem

In this section it will be shown how Specker's Theorem follows from Theorem 2.2.2 (the statement of Specker's Theorem is given in section 1.5).

In Theorem 2.2.2 set D = {0, 1}, $\Phi = \Sigma$, a = 0 and let f be as described. Then by Theorem 2.2.2, we obtain that for an appropriate choice of n, we can find a subset X of the arguments of f with $|X| = m$ and such that f_0^X is either a constant or represented by

$$\psi(F_1,\ldots,F_r)$$

where $\psi:\{0, 1\}^r \rightarrow \{0, 1\}$ and F_i for $1 \leq i \leq r$ is either a standard or reverse homogeneous e_m-component over Σ^0 with the I^0-property and constant input operators. The value of r is bounded as described in Theorem 2.2.2.

We now analyze the various functions that can be represented by e-components with these restrictions. First note that Σ^0 consists of all Boolean binary operators; furthermore, if $f(x,z) \in \Sigma^0$, then both x and z are free (because every Boolean binary operator can be represented over Σ in such a way that each variable appears only once), and thus there are no restrictions on the use of operators in the e-components we encounter.

All possible graphs $\Gamma(\varphi)$ for $\varphi \in \Sigma^0$ are shown in Fig. 2.7. The ones that satisfy the I^0-property are starred. The functions obtained by choosing a value for the constant input operator for the starred graphs are shown in Table 2.1. In general, this function is either $b_0 \oplus b_1 \cdot \pi$ where $\pi = \prod_{i=1}^{m} (1 \oplus x_i)$ or $c_0 \oplus c_1 \cdot \sigma$ where $\sigma = \bigoplus_{i=1}^{m} x_i$ for some values of b_0, b_1 or c_0, c_1. Now, taking into consideration the fact that $\pi \cdot \sigma = 0$, and that every $\psi: \{0, 1\}^h \rightarrow \{0, 1\}$ can be uniquely expressed as a Boolean polynomial

$$\bigoplus_{i=0}^{2^h-1} c_i \cdot M_i$$

where $c_i \in \{0, 1\}$ and M_i is the monomial (of degree one in each variable) in those among $x_1,\ldots,x_h$ corresponding to nonzero bits of the binary representation of i (see Lemma 4.5) we obtain the first part of Specker's Theorem.

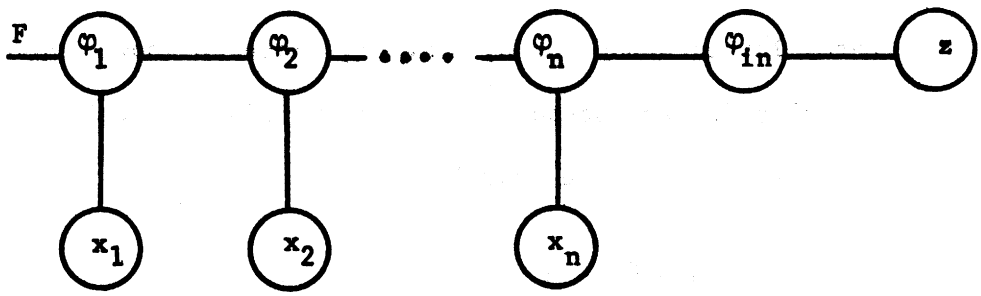
The second part of Specker's Theorem could be obtained directly at this point; however, we will derive a generalization of it in Example 3.1.3, and thus omit it here.

It must be pointed out that our derivation of Specker's Theorem results in a slightly larger bound for n; however, since no known application requires a specific value for the bound, this is immaterial. Specker's bound (see [Ho68]) is obtained from the function

$$\mu(m,0) = m$$
$$\mu(m,k) = 4^{(k+1)^{6k\mu(m,k-1)}} \cdot \mu(\mu(m,k-1),k-1)$$

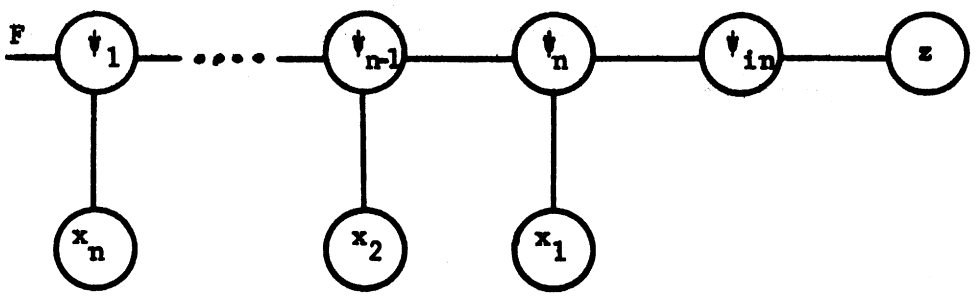
by setting $\eta_S(m,c) = 2\mu(m,2c)$. Our bound is slightly larger due to the additional processing implicit in the application of Proposition 2.1.7 and Lemma 2.1.8. However, μ resembles η_7 and this function by far contributes the most to η_5; thus, we can state that the bounds are approximately equal.

Finally, let us note the fact that Theorem 2.2.2 allows us immediately to amplify Specker's Theorem. Namely, the statement of the theorem involves the basis consisting of all binary Boolean operators. However, the proof of Theorem 2.2.2 works for bases consisting of operators of an arbitrary number of arguments.



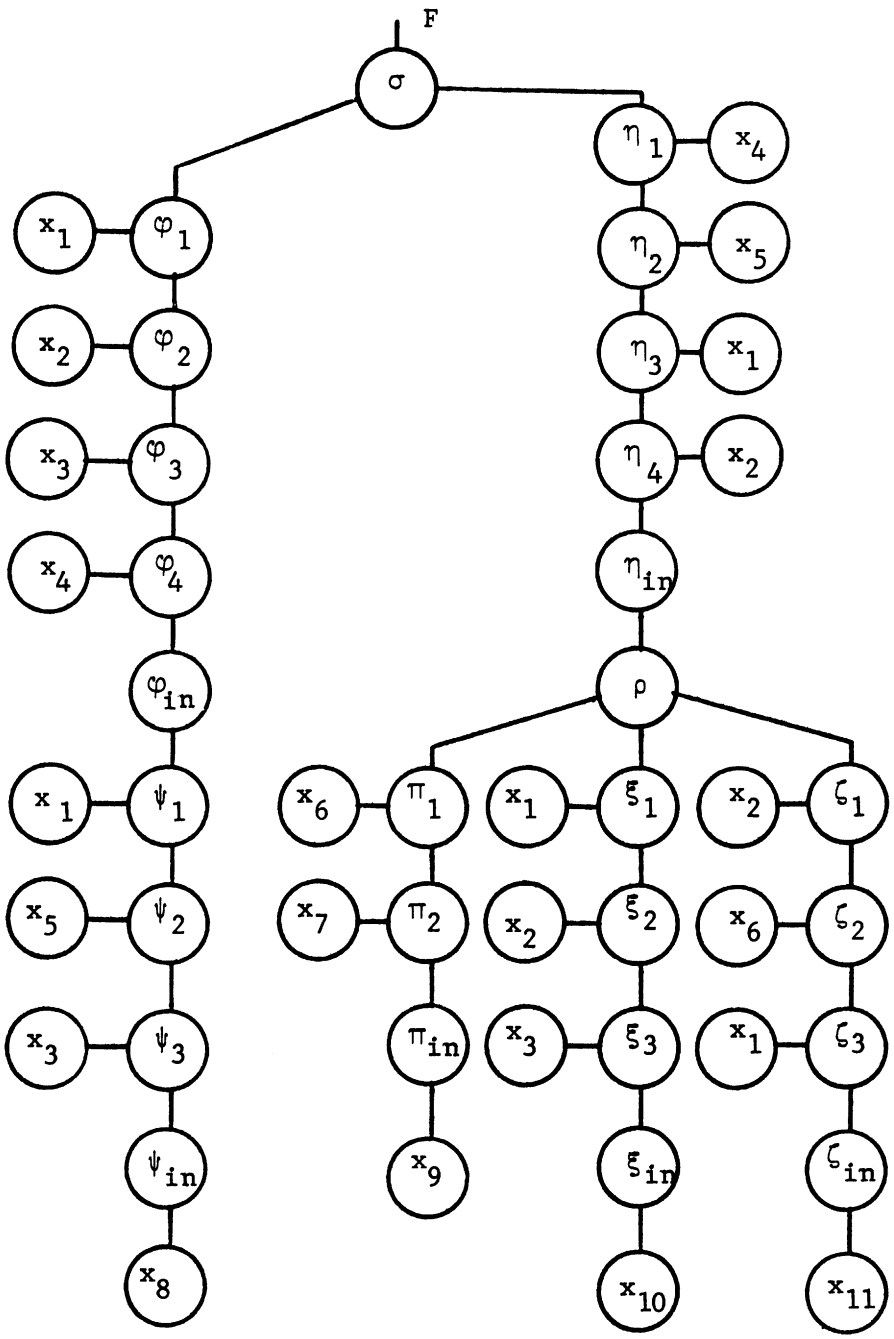
T(F) where F is an e_n-component

Fig. 2.1



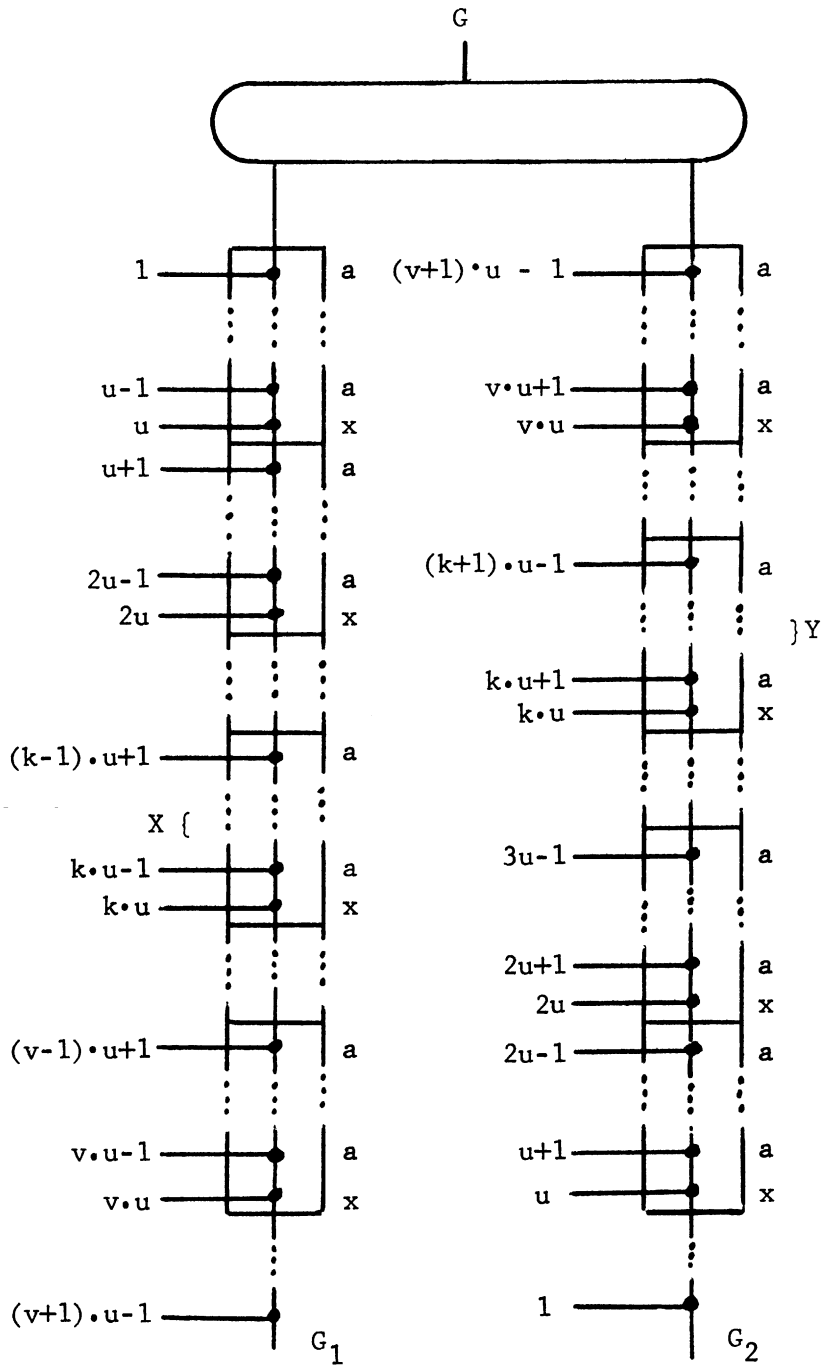
T(F) where F is a reverse component of an e-complex

Fig. 2.2



F is an SPCeC

Fig. 2.3



$X = \varphi_{((k-1)\cdot u+1,\, k\cdot u)}$

$Y = \psi_{((v-k)\cdot u+1,\, (v-k+1)\cdot u)}$

Illustration of the HP procedure (I)

Fig. 2.4

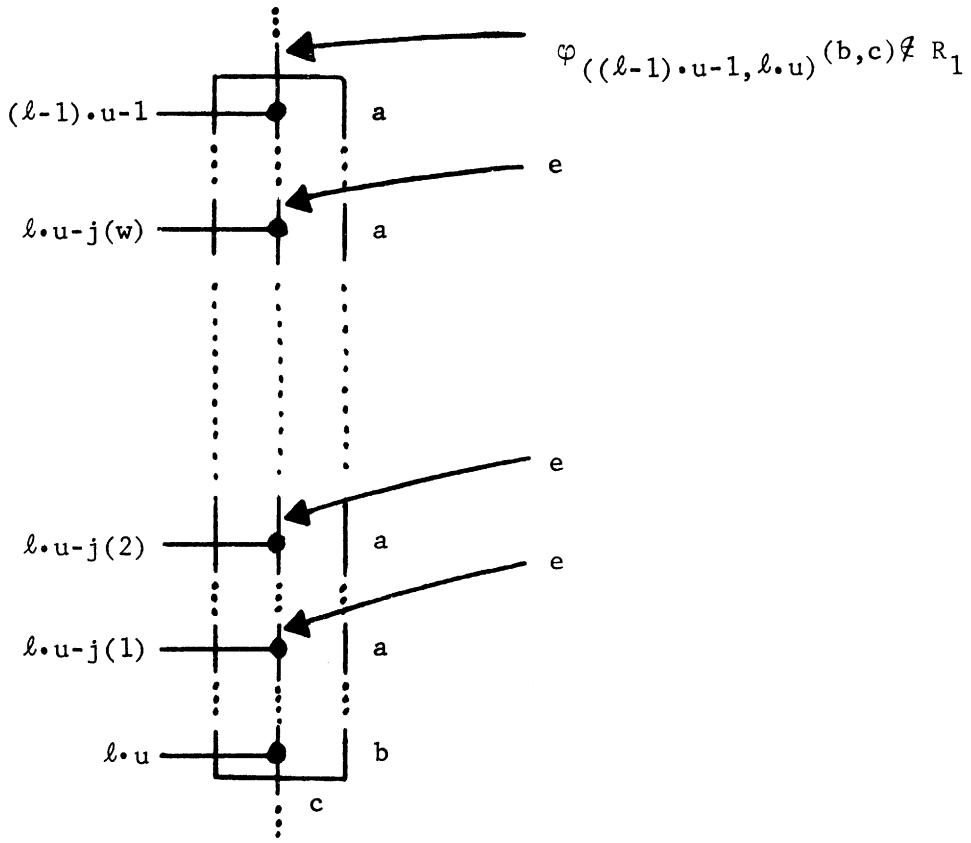
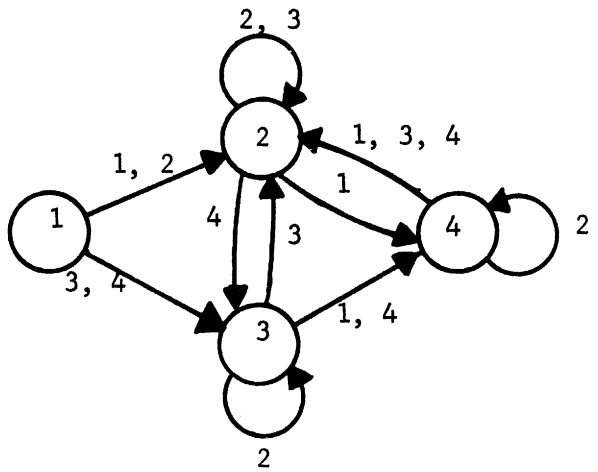


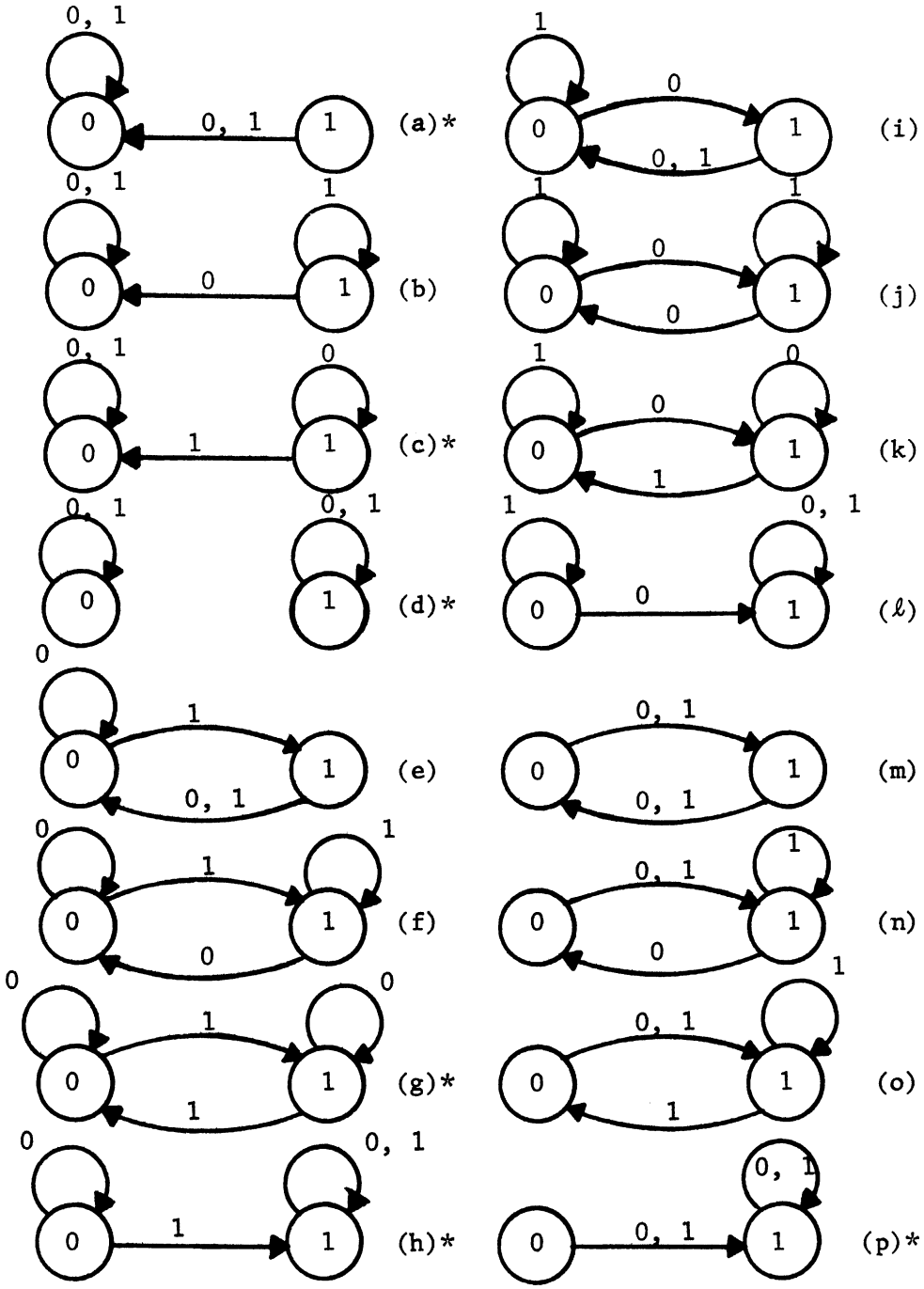
Illustration of the HP procedure (II)

Fig. 2.5



$\Gamma(\varphi)$ for an operator φ with the $I^2_{\{2,3,4\}}$-property

Fig. 2.6



The graphs of all Boolean binary operators

Fig. 2.7

$\Gamma(\varphi)$	$\varphi_{in}(y)$	Function
a	0	0
a	1	0
c	0	0
c	1	$\bigsqcap_{i=1}^{m}(1 \oplus x_i)$
d	0	0
d	1	1
g	0	$\bigoplus_{i=1}^{m} x_i$
g	1	$1 \oplus \bigoplus_{i=1}^{m} x_i$
h	0	$1 \oplus \bigsqcap_{i=1}^{m} (1 \oplus x_i)$
h	1	1
p	0	1
p	1	1

Table of functions that can be represented by e-components with the I^0-property (see Fig. 2.7) and constant input operators if D = {0, 1}

Table 2.1

CHAPTER THREE

APPLICATIONS OF THE GENERALIZED SPECKER THEOREM

The principal results obtained previously [Ho68, Ho70] by the use of Specker's Theorem are

3.0.1

A new proof that the Boolean function $\bigoplus_{i=1}^{n} x_i$ is of nonlinear length over Π[†]. This is accomplished as follows. First note that the restriction of the mod 2 sum of n variables obtained by setting certain variables to 0 is again a mod 2 sum (but of a smaller number of variables). Now apply Specker's Theorem (see 1.5). Suppose $\bigoplus_{i=1}^{n} x_i$ is of linear length over Π. Choose n large enough to obtain m = 3. The theorem states that for this particular bases $c_2 = 0$ in (1.5.1). However, it can be checked that in this case no choice of c_0 and c_1 will yield the mod 2 sum of three variables. A contradiction.

3.0.2

The function $f: \{0, 1\}^n \to \{0, 1\}$ defined by f = 1 if and only if $\sum_{i=1}^{n} x_i \equiv 0$ mod 3 is of nonlinear length over Σ. We proceed similarly as before. Assume it is of linear length. Apply Specker's Theorem with n sufficiently large to

[†]Of course the results of Subbotovskaya and Khrapchenko are stronger for this particular example.

obtain m = 3. If we replace x_1, x_2, x_3 in (1.5.1) once by 1, 0, 0, and another time by 1, 1, 1, then the value of (1.5.1) remains unchanged. However, the value of f (with all variables except x_1, x_2, x_3 replaced by the constant 0) is different on these two assignments. Again a contradiction.

Both of these results were derived by Hodes and Specker in [Ho68].

We might note that the technique of 3.0.2 can easily be generalized to counting mod k where k is an arbitrary integer (see (1,5.2)).

3.0.3

Certain geometric predicates (see [Mi69]), in particular the connectivity predicate, are of nonlinear length if expressed with binary Boolean operators (this result was obtained by Hodes in [H070]). We will not discuss this in greater detail now since this technique will be treated later in 3.2.

In this chapter we will use Theorem 2.2.2 to generalize all these results.

3.1 Counting mod p

Consider the function $\{0, 1\}^n \to \{0, 1\}$

$$f_n^p(x_1,\ldots,x_n) = \begin{cases} 1 \text{ if } \sum_{i=1}^{n} x_i \equiv 0 \bmod p \\ 0 \text{ otherwise} \end{cases}$$

then

3.1.1 Theorem

If p is a prime, if $|D| < p$, then f_n^p is not of linear length over an arbitrary basis.

Proof

Suppose the statement of the theorem is not true. That is, there exists a prime p, a finite set D such that $|D| < p$, a basis Φ of operators on D, and $L(f_n^p, \Phi) \leq c \cdot n$ for some constant c.

First note that if X is a subset of the arguments of f_n^p and $|X| = m$, then $(f_n^p)_0^X = f_m^p$. We can now apply Theorem 2.2.2. For an arbitrary m, if n is sufficiently large, there exists a subset X of the arguments of f_n^p with $|X| = m$ and such that $(f_n^p)_0^X$ is represented by a homogeneous e_m-complex F (over Φ^0 and with the I^0-property) with X as the set of lateral variables. In addition, since f_n^p is a Boolean function, the lateral variables of F are restricted to {0, 1}.

Consider now any component F_i of F and $\Gamma(\varphi_i)$ where φ_i is the internal operator of F_i. Since φ_i has the I^0-property, $\Gamma(\varphi_i)$ has the general appearance of Fig. 3.1. F_i is determined by φ_i and the constant input operator a_i. Now let $m \geq d$ and consider the sequence (s_j) for $0 \leq j \leq m$ where $s_0 = a_i$ and $s_j = \varphi_i(11\ldots11,a_i)$ for $j > 0$. Let $s_{k(i)}$ be the first element in the sequence (with j 1's) that is repeated at some later point; in fact, let k(i) be the position of the first occurrence of this element. Let $k(i)+\ell(i)$ be the position of the second occurrence of this same element. Then we shall call k(i) the prefix of F_i while $\ell(i)$ will be called the period of F_i.

Clearly, if F_i is a standard (reverse) component, then $\varphi_i(x_1x_2\ldots x_{m-k}\,11\ldots1,a_i)$ $(\varphi_i(x_mx_{m-1}\ldots x_{k+1}11\ldots1,a_i))$ where $k \geq k(i)$ is a function of the number of 1's among $x_1, x_2,\ldots,x_{m-k}$ $(x_{k+1},\ldots,x_m)$ mod $\ell(i)$.

Thus,

> If we set $Y = \{x_{k_1+1},\ldots,x_{m-k_2}\}$ and choose $k_1(k_2)$ to exceed or equal the prefixes of all the reverse (standard) components of F, then F_1^Y represents a function of the number of 1's among the variables of Y mod $\ell cm(\ell(1),\ldots,\ell(r))$. (3.1.1)

On the other hand, by the initial assumption, F_1^Y is a function of the number of 1's among the variables of Y mod p; this results in a contradiction since $d < p$.

On the basis of (3.1.1) we can obtain the following

3.1.2 Theorem

Let D be an arbitrary domain, Φ is a certain basis, and p is an arbitrary integer > 1. If Φ^0 is such that any e-component over Φ^0 with the I^0-property and constant input operator has period one, then f_n^p is of nonlinear length over Φ.

3.1.3 Example

This is an example of a basis satisfying the hypothesis of Theorem 3.1.2.

Consider an arbitrary domain $D = \{0,1,\ldots,d-1\}$. Then a complete basis for D is $\psi_D = \{\min(x,y), \max(x,y),0,1,\ldots,d-1, e_0(x),\ldots,e_{d-1}(x)\}$ where min and max are defined in the usual way, 0,...,d-1 are the constants, and

$$e_i(x) = \begin{cases} d-1 \text{ if } x = i \\ \\ 0 \text{ otherwise} \end{cases}$$

(Note that $\psi_{\{0,1\}} = \{\wedge,\vee,0,1,-,id\}$; thus, a result on the nonlinearity of the length of a certain function over $\psi_{\{0,1\}}$ is also a result on the nonlinearity of the same function over Π; in particular, applying Theorem 3.1.2 to $\psi_{\{0,1\}}$, we obtain (2) of Specker's Theorem.)

ψ_D is interesting because it gives rise to an analog of the disjunctive normal form for arbitrary D: Consider the table for an arbitrary function $f: D^n \to D$. Then

$$f = \max_{i=0}^{d^n-1} (M_i)$$

where M_i equals 0 if the current assignment is not the ith assignment and the value of the function at the ith assignment otherwise. M_i is represented as follows

$$M_i = \min(e_{a(i,1)}(x_1),\ldots,e_{a(i,n)}(x_n),f_i)$$

where $a(i,j)$ is the jth component of the ith assignment.

We claim that ψ_D satisfies the hypothesis of Theorem 3.1.2. Note that $\psi_D^a = \psi_D^b$ for all $a, b \in D$ because ψ_D contains all the constants. Therefore, we will write simply ψ_D^*.

Given $\varphi \in \underline{\psi}_D^*$ with the I^0-property, the statement that there exists $b \in D$ such that the homogeneous e-component with internal operator f and b for its input operator has period ℓ is equivalent to saying that there exists a subset $L \subseteq D$ with $|L| = \ell$ and $\varphi(0,z)\ \ L$ is the identity (id_L) while $\varphi(1,z)\ \ L$ is the permutation with cycle length ℓ (p_ℓ).

We contend that for any $L \subseteq D$, $\varphi(x,z) \in \underline{\psi}_D^*$ and $c, e \in D$, if $\varphi(c,z)\ \ L$

and $\varphi(e,z)$ L are 1-1, then $\varphi(c,z)$ L = $\varphi(e,z)$ L. Since $id_L \neq p_\ell$ if $\ell > 1$, this will establish the original claim.

This can be proved by induction on the depth δ_φ of the distinguished variable z in the formula F that represents φ (since there may be many such formulas, let F be one of the formulas where the depth of z is minimal.

If $\delta_\varphi = 1$ then either F = max(F',z), or F = min(F'',z). Assume the first case (the second can be argued similarly). By definition of $\underline{\Psi}^*_D$, F' contains only the variable x. If we replace x by c, F' represents a constant $c' \in D$. Now if $c' \leq L$, then $\varphi(c,z)$ L is the identity, otherwise it is not 1-1.

If $\delta_\varphi > 1$, then either $\varphi(x,z) = e_i(\varphi'(x,z))$ where $\varphi' \in \underline{\Psi}^*_D$ and $\delta_{\varphi'} < \delta_\varphi$ or $\varphi(x,z) = \varphi^*(x,\varphi'''(x,z))$ where $\varphi'', \varphi''' \in \underline{\Psi}^*_D$ and $\delta_{\varphi''}, \delta_{\varphi'''} < \delta_\varphi$ (to see this, think of F). In any case φ', φ'', and φ''' satisfy the inductive hypothesis, and we are done.

Note that Theorem 3.1.1 and 3.1.2 hold with f^p_n replaced by the function $g^p_n: D^n \to \{0,1\}$ given by $\sum_{i=1}^{n} x_i = 0 \bmod p$ since $g^p_n \upharpoonright \{0,1\}^n = f^p_n$.

3.2 Connectivity

The connectivity predicate was already discussed in 1.6. It attacted considerable attention after Minsky and Papert [Mi69] succeeded in obtaining interesting results on the complexity of perceptrons that represent the connectivity predicate. Works that follows [Mi69] and that treat specifically the representation of the connectivity predicate by finite operators are, e.g., [H070], [Mi71], and [Vi70].

Minsky and Papert describe a circuit for computing the connectivity predicate of depth (of the order of) $(\log_2 n)^2$ which on intuitive grounds seems minimal. This circuit translates into a formula of nonpolynomial length. Thus, the connectivity predicate seems to be a good benchmark for testing estimation methods for the complexity of functions (i.e., any appropriately general method which is presumed able to give estimates for length up to $f(n) < n^{\log_{2n}}$ should declare the connectivity predicate complex).

Consider a set of n^2 variables $\{x_{ij}\}$ for $1 \leq i, j \leq n$; then the connectivity predicate is the function c_n: $\{0, 1\}^{n^2} \rightarrow \{0, 1\}$ defined as follows: (we will not give a formal definition since the formalization is obvious) Given a specific assignment to the variables, consider it as a square array of 0's and 1's. Then $c_n = 1$ on the empty pattern (i.e., consisting of all 0's), or if the 1's form a connected pattern. By "connected" we mean that any two 1's can be linked by a sequence of adjacent 1's (two 1's, corresponding to the variables x_{ij} and $x_{k\ell}$ are adjacent if $|i-k| + |j-\ell| = 1$). For example, the pattern in Fig. 3.2 is connected.

The general approach used here to obtain an estimate for the length of c_n is to consider reductions of c_n.

Given an arbitrary function f, a function g will be called a k-reduction of f if g is obtained from f by replacing each argument of f by a function with at most k arguments.

Suppose we want to prove that f_i of $i = 1,2,\ldots$ arguments is of nonlinear length (over the basis Φ). Assume there exists a k-reduction g_i of f_i such that the number of arguments m in g_n is $m \geq \alpha \cdot n$ for some constant $0 < \alpha \leq 1$. Assume $L(f_n, \Phi) \leq c \cdot n$ for some constant c. That is, there exists a formula

F_n for f_n and $L(F_n) \leq c \cdot n$. Since the length of any function of k arguments is bounded by $L(k,\Phi)$, we obtain

$$L(g_n,\Phi) \leq c \cdot L(k,\Phi) \cdot n$$

by making substitutions for variables in F_n.

If $\{g_i\}$ is rearranged (and renumbered) in the order of increasing number of arguments, and all but one functions with the same number of arguments are deleted, then we obtain

$$L(g_m,\Phi) \leq c' \cdot m$$

for some constant c'. Finally, if we can prove (e.g., by applying Theorem 2.2.2) that g_m is nonlinear, we obtain a contradiction, and we are done.

Hodes [Ho70] shows the nonlinearity of the length of c_n over Σ by reducing c_n to the function

$$\bigvee_{i=1}^{m} \left(\bigwedge_{j \neq i} y_j \right) \wedge y_i$$

(i.e., exactly one variable is 1) which can then be proved to be nonlinear using Specker's Theorem. Unfortunately, this reduction does not work for domains with more than two elements because this function is linear over an appropriate basis in such domains. However, another approach works, and we can state

3.2.1 Theorem

Regardless of the size of D and the nature of Φ, c_n is of nonlinear length (i.e., $L(c_n,\Phi) \leq c \cdot n^2$ is not true for any D, Φ, and constant c).

Proof

Minsky and Papert [Mi69] succeed in reducing c_n to counting mod 2 by exhibiting a contact network such that its connectivity depends on the number mod 2 of contact variables equal to 1, and then by simulating this network on the square array of variables (they call it the "retina"). We shall proceed similarly.

c_t is reduced to the function s_n^p: $\{0, 1\}^n \to \{0, 1\}$ (for an appropriate t) defined as follows:

$$s_n^p = \begin{cases} 1 \text{ if } \geq \text{ arguments are equal to 1} \\ 0 \text{ otherwise} \end{cases}$$

s_n^p, can be represented by the connectivity of a contact network S_n^p. S_n^p is shown in Fig. 3.3a. It has p contact arms for each variable y_i. The 0 value of y_i corresponds to the upward position of the corresponding arms while the 1 value of y_i corresponds to the downward position. The contact arm of S_n^p are arranged in p rows (n arms in each row). Whenever an arm for y_i is in the upward position, it is connected to an arm for y_{i+1} in the same row; if the arm for y_i is in the downward position, it is connected to an arm for y_{i+1} in the next row. Thus, it may be easily checked, point A_1 is connected to B_{p+1} if and only if at least p among $y_1,\ldots,y_n$ are 1. It may also be verified that in this case all the contact arms in the network are connected either to A_0 or to B_{p+1}, and since these two are connected together, the whole network is connected.

S_n^p in turn can be simulated by a rectangular array R_n^p of 0's and 1's where certain positions are constant and others depend on the y_i's (see Fig. 3.3b). The size of R_n^p is $(3(p+1)+1).(3n+p-1)$.

We now show that s_n^p is a 1-reduction of c_t for some t. This is done by cutting R_n^p into smaller rectangular pieces along vertical lines. The first piece is of length $(\ell-1)\cdot q$ where $q = 3(p+1)+1$ and ℓ will be defined later, the second through ℓ-1st piece is of length $(\ell-2)\cdot q$, while the ℓth, last, piece is of length between 1 and $(\ell-1)\cdot q$. These pieces are then arranged into an $\ell\cdot q \times \ell\cdot q$ square pattern T_n^p as shown in Fig. 3.4 (the arrangement depends on the parity of ℓ). Corresponding rows of adjacent pieces are connected by $\supset$- or $\subset$- shaped patterns of 0's (in the case when one of the positions along the cut at the row in question is 0) or 1's. The unused positions of T_n^p (corresponding to the case when the last piece is not of the maximal length) are replaced by 0's.

t is set to $\ell\cdot q$ and the variable x_{ij} in c_t is replaced by the corresponding position in T_n^p (one among 0, 1, y_i, or $\bar{y}_i$). Obviously, the function obtained by this replacement is s_n^p. If $\frac{3n+p-1}{q} \geq 1$, ℓ is obtained as $\lceil x \rceil$ where x is the positive solution of

$$x^2-2(x-1) = \frac{3n+p-1}{q}$$

We also have that $n \approx (1/3q)t$, and, thus (by the reasoning outline previously), if c_t is linear so is s_n^p.

With the assumption that s_n^p is linear, we apply Theorem 2.2.2 with a = 0. In this way we obtain that $(s_n^p)_0^Z$ where Z is a certain subset of the arguments of s_n^p of size m is represented by an e_m-complex F (with the requisite restrictions) where m is arbitrary. Note that $(s_n^p)_0^Z = s_m^p$.

As noted in 3.1, if a sufficiently large number of variables at the beginning and end of the lateral sequence of F is replaced by 1, then F with this substitution represents a function of the number of 1's among the remaining variables mod the lcm of a set of integers $\leq d$. The number of variables that have to be set of 1 is $u \leq 2(d-1)$ (at most d-1 at each end of the lateral sequence). Thus we obtain a representation of the function s^{p-u}_{m-u} if $p \geq u$. If $p \geq 2(d-1)+2$, we obtain a function s^i_j for $i \geq 2$. However, it is clear that s^i_j is not a function of the number of 1's mod k for any integer k. Thus, we have arrived at a contradiction, and, hence, c_n is of nonlinear length over any basis Φ. □

3.3 The Length of Symmetric Functions†

As we have seen in the previous examples, Theorem 2.2.2 has been applied only to functions that are either symmetric or that can be reduced to symmetric functions. While we know of no formal statement that can be proved and that asserts that this indeed exhausts the applicability of Theorem 2.2.2, it intuitively seems probable.

In this section we will discuss several bounds on the length of symmetric functions (both specific functions and all symmetric functions). Recall that in 1.4 we have already mentioned several such bounds (Subbotovskaya, Khrapchenko).

---

†All of the results in this section were suggested by A. R. Meyer.

Does Theorem 2.2.2 (or Specker's Theorem) give us any information on the length of the functions investigated? Hodes and Specker do not treat this subject, and, in fact, the bound that can be obtained is very weak; however, we do mention it for the sake of completeness.

In an application of Theorem 2.2.2 (or Specker's Theorem) to a certain function f, we proceed with the assumption that $L(f,\Phi) \leq c \cdot n$. To apply Theorem 2.2.2 we must have $n \geq \eta_5(m,c)$ where m is a sufficiently large number to obtain a contradiction. However, m does not depend on c. Thus, n depends only on c and m is assumed constant.

Consider now c as a function of n. We ask what is the maximal value $\bar{c}$ for c(n) for which Theorem 2.2.2 can be applied (and a statement contradicting $L(f,\Phi) \leq c \cdot n$ obtained). $\bar{c}(n) \cdot n$ is then a lower bound for $L(f,\Phi)$. Due to (2.2.1) $\bar{c}$ grows slower than $(1/4) \cdot ht\ (p,n)$ where ht (p,n) = maximal x such that $n \geq iexp\ (p,x)$. Then we have

$$\bar{c}(n) \cdot n \leq (1/4) \cdot ht\ (p,n) \cdot n \tag{3.3.1}$$

for an arbitrary constant $b > 1$ and for sufficiently large n.

This bound seems unrealistically low, and it is useful to compare it with known bounds for length for the particular function f_n^3 over some bases consisting of Boolean operators (we will suppress the subscript n).

It has already been established that f^3 is of nonlinear length if D = {0, 1} (see 3.1). We introduce the following notation: $f^3 = f^{3,0}$, $f^{3,1}$, and $f^{3,2}$ stand for $\sum_{i=1}^{n} x_i \equiv 0$, 1, and 2 mod 3 respectively. We will represent $f^{3,0}$, $f^{3,1}$, $f^{3,2}$ by the formulas F^0, F^1, and F^2 respectively. F^0 is obtained by the following recursive relation

$$F^0(x) = F^0(Y) \wedge F^0(Z) \vee F^1(y) \wedge F^2(Z) \vee F^2(Y) \wedge F^1(Z) \qquad (3.3.2)$$

(If X is the singleton $\{x\}$, then $F^0(X) = \bar{x}$)

$$L(F^0(X)) = L(F^0(Y))+L(F^0(Z))+L(F^1(Y))+L(F^2(Z))+L(F^2(Y))+L(F^1(Z))$$

Similar identities can be obtained for $F^1(X)$ and $F^2(X)$. When these identities are used recursively, we obtain

$$O(L(f^3,\Phi)) \leq n^{\log_2 6} \approx n^{2.6} \qquad (3.3.3)$$

an exact description how we obtain a bound of the form (3.3.3) from a recursive relation similar to (3.3.2), see [Ya54].

This upper bound can be further reduced by using multiargument operators. Let γ: $\{0,1,2\}^{\ell} \to \{0,1,2\}$ be the operator $\sum_{i=1}^{R} y_i \mod 3$. Then f^3 can be represented by a formula G which uses γ recursively (i.e., the arguments of f^3 are repeatedly divided by ℓ together with an outermost decoding operator $\{0,1,2\} \to \{0,1\}$. G is of linear length. If we use $D = \{0,1\}$, γ can be encoded by two operators γ' and γ'', and G translates into a formula such that

$$O(L(G)) = (2k)^{\log_k n} = n^{1 + \frac{1}{\log_2 k}} \qquad (3.3.4)$$

Thus, as ℓ increases, the upper bound for $L(f^3,\Phi)$ (where $\gamma \quad \Phi$) approaches $c \cdot n$. However, the gap between this bound and (3.3.1) is still huge. But, the important thing to note is that any theorem that retains the same broad assumption (bases with an arbitrary number of operators) as Theorem 2.2.2 cannot yield a better bound for f^3 than (3.3.4).

Another example of a function that is nonlinear in length (over all Boolean binary operators) by Specker's Theorem is f^4. However, it too has a relatively short representation (the previous and this example show that Theorem 2.2.2 is a sensitive tool for deriving the nonlinearity of functions; i.e., it can be used on functions that are only "slightly" nonlinear).

A representation for f^4 with Boolean operators is obtained by dividing the arguments of f^4 into disjoint (nonempty) pieces Y and Z, and adding the bineary representations of $f^4(Y)$ and $f^4(Z)$. Let the binary representations of $f^4(X)$ be given by the formulas F'(X) and F"(X), obtained by the following recursive relations

$$F"(X) = F"(Y) \barwedge F"(Z)$$

$$F'(X) = F'(Y) \oplus F'(Z) \oplus F"(Y) \wedge F"(Z)$$

(If X is a singleton F"(x) = x and F'(x) = 0)

Consequently,

$$L(F"(X)) = L(F"(Y))+L(F"(Z))$$

$$L(F'(X)) = L(F'(Y))+L(F'(Z)+L(F"(Y))+L(F'(Z))$$

By choosing Y and Z always as equal as possible, we obtain

$$L(F"(X)) = n$$

$$O(L(F'(X))) = n \cdot \log_2 n$$

Since $f^4(X)$ is represented by $\overline{F'(X)} \wedge \overline{F"(X)}$ $n \cdot \log_2 n$ is also a bound for $O(L(f_n^{\Phi})$ where Θ, $\wedge$ Φ.

We now turn our attention to an upper bound for the length of all symmetric functions.

Note that a symmetric function g: $D^n \rightarrow D$ where $D = \{0,1,\ldots,d-1\}$ depends exclusively on $N_1,\ldots,N_{d-1}$ where N_i is the number of variables equal to i. It can be represented, e.g., as

$$g = \max(M_{n(1)\cdots, n(d-1)})$$

where $M_{n(1)\ldots, n(d-1)}$ equals Ψ if $N_i = n(i)$ and is 0 otherwise. The number of combinations of $n(1),\ldots,n(d-1)$ is a polynomial in n -- $\binom{n+d-1}{d-1}$ -- and the max function can also be represented in polynomial length, regardless of the basis Φ. The latter fact is established by representing max using the two-argument max recursively. Thus, if $M_{n(1),\ldots,n(d-1)}$ were polynomial, g would also be.

M. J. Fischer and A. R. Meyer discovered that $M_{n(1),\ldots,n(d-1)}$ can, indeed, be represented in polynomial length by using a special code for integers described by Avizienis [Av69].

We will illustrate the construction on Boolean symmetric functions. It will be seen that if the basis of operators is appropriately chosen, the length of an arbitrary symmetric function is bounded above by a polynomial of a surprisingly low degree.

The Avizienis code is a redundant positional representation of integers to an arbitrary base $b > 2$. We describe it for $b = 3$.

An integer n is represented by all possible $\lceil \log_3 n \rceil$ - tuples

$$a_{\lceil \log_3 n \rceil},\ldots,a_1$$

where $a_i \in \{-2,-1,0,1,2\}$ for $1 \leq i \leq \lceil \log_3 n \rceil$ and

$$\sum_{i=1}^{\lceil \log_3 n \rceil} a_i 3^{i-1} = n$$

The property that is exploited is that there are no long carry's in addition. Thus, if we want to add two Avizienis coded integers $a = a_k a_{k-1} \cdots a_1$ and $b = b_k b_{k-1} \cdots b_1$, we can do it in two steps using the following

3.3.1 Algorithm (Avizienis)

(1) Find the carry c and intermediate sum r such that

$$a_i + b_i = 3c_i + r_i$$

where $a_i, b_i \in \{-2,-1,0,1,2\}$ and $c_i, r_i \in \{-1,0,1\}$.

(2) Compute the sum s according to

$$s_i = r_i + c_{i-1}$$

Let us estimate the length of the formula representing any ternary place in the Avizienis representation of N_1 for $X = \{x_1,\ldots,x_n\}$.

Again let $X = Y \cup Z$ and $Y \cap Z = \phi$. $r_i(X)$ and $c_i(X)$ can be represented as

$$R_i(X) = \rho(R_i(Y),R_i(Z),C_{i-1}(Y),C_{i-1}(Z))$$

and

$$C_i(X) = \chi(R_i(Y),R_i(Z),C_{i-1}(Y),C_{i-1}(Z))$$

If X is a singleton, r_i and c_i are 0 if $i > 1$, or 1 and 0 respectively if $i = 1$. ρ and χ are certain operators $\{-1,0,1\}^4 \rightarrow \{-1,0,1\}$ which can be obtained from the definition of Algorithm 3.3.1. Strictly speaking, the

domain used here is not permitted in our definition of finite functions; however, the difference is merely one of coding. Thus,

$$L(R_i(X)) = L(R_i(Y)+L(R_i(Z)+L(C_{i-1}(Y))+L(C_{i-1}(Z))$$

and

$$L(C_i(X)) = L(R_i(Y))+L(R_i(Z))+L(C_{i-1}(Y))+L(C_{i-1}(Z))$$

If we use these relations recursively and always make Y and Z as equal as possible, we obtain

$$O(L(R_i)),\ O(L(C_i)) \leq n^2$$

for $1 \leq i \leq \lceil \log_3 n \rceil$. If $D = \{0,1\}$, we need two bits to encode r_i and c_i. Therefore, using certain operators ρ', ρ'', χ', and χ'' to encode ρ and χ, we can encode $R_i(X)$ and $C_i(X)$ and combine them into a $\{0,1\}$-formula A_i representing the ith ternary place of the Avizienis representation of N_1. We have

$$O(L(A_i)) \leq n^3 \tag{3.3.5}$$

Let there be given a positive Avizienis coded number $a = a_p a_{p-1} \ldots a_i$. We desire to convert it into its binary equivalent $b = b_q b_{q-1} \ldots b_1$. Let $U \subseteq \{a_p, \ldots, a_1\}$. Then we define $b_i(U) = 1$ if and only if the ith bit of $\sum_{\alpha_i \in U} a_i 3^{i-1}$ is 1. Note that even if a is positive, $b_i(U)$ may be negative for some i and U. This is further discussed below. For the moment we assume that $b_i(U)$ is always positive. We can then again compute $b_i(a_p \ldots a_1)$ by a recursive method. Let $U = V \cup W$, $V \cap W = \phi$. Then $b_i(U)$ is the ith bit of the sum of $b_q(V) \ldots b_1(V)$ and $b_q(W) \ldots b_1(W)$.

$b_i(U)$ is represented by the formula B_i

$$B_i(U) = \beta(B_i(V),B_i(W),G_{i-1}(U))$$

where G_i represents the carry from the ith place;

$$G_i(Y) = \gamma(B_i(V),B_i(W),G_{i-1}(U))$$

G_0 represents the constant 0 and β and γ are certain Boolean operators. Then we obtain

$$L(B_i(U)) \approx \sum_{j=1}^{i} L(B_j(V))+L(B_j(W))$$

If a is the Avizienis representation of a number $\leq n$ then $p = \lceil \log_3 n \rceil$ and $q = \lceil \log_2 n \rceil$. Thus, we obtain the following bound for $L(B_i(a))$

$$O(L(B_i(a))) \leq (2q)^{\log_2 \log_3 n} \approx$$

$$(2\log_2 n)^{\log_2 \log_3 n}$$

$$\frac{(1+\log_2 \log_2 n)\log_2 \log_3 n}{\log_2 n}$$

$$\approx n \qquad (3.3.6)$$

Note that (3.3.6) means that $O(L(B_i(a))) \leq n^{\varepsilon(n)}$ for $1 \leq i \leq \lceil \log_2 n \rceil$ where $\varepsilon \to 0$ as $n \to \infty$.

It has already been remarked that b(U) need not be positive. Thus b(U) must be treated as a signed number. If we use the 1's complement representation and the en-around carry technique (see, e.g., [Gr59]), addition can be performed as follows. Let $b(U,g_0)=G(V)+b(W)+g_0$ where g_0 is either 0

or 1 and let g_1 denote the carry from the highest position of b(U,0). Then b(U) = b(U,0) if $g_1 = 0$ and b(U,1) if $g_1 = 1$. This means that in (3.3.4) we obtain $(\ell.q)^{exp}$ for some constant ℓ instead of $(2q)^{exp}$ where exp has the value given above.

If a_j for $1 \le j \le \lceil \log_3 n \rceil$ in B_i for $1 \le i \le \lceil \log_2 n \rceil$ is replaced with A_j, we obtain formulas $F_i(X)$ representing N_1 in binary form. Combining (3.3.5) and (3.3.6) we obtain

$$O(L(F_i(X), \ldots)) \le n^{3+\epsilon(N)}$$

where $\epsilon(n) \to 0$ as $n \to \infty$.

To obtain the desired representation of an arbitrary Boolean symmetric function, we proceed as follows: Consider the formula S_i defined inductively

$$S_1 = x_{10} \vee x_{11}$$

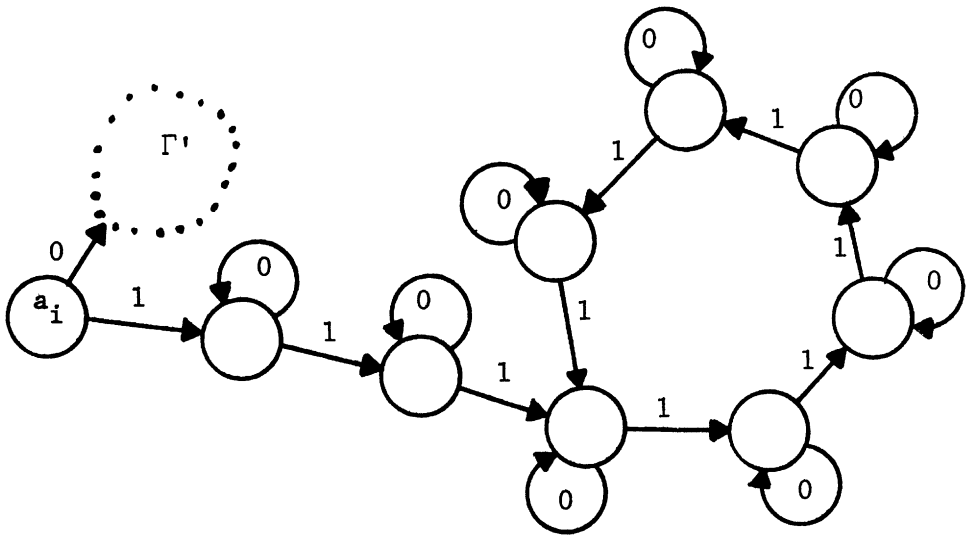
$$S_{i+1} = x_{i0} \wedge S_{i-1} \vee x_{i1} \wedge S_{i-1} \qquad (3.3.7)$$

Take $S_{\lceil \log_2 n \rceil}$ and replace x_{i0} and x_{i1} by $\bar{F}_i$ and F_i respectively. It is easily seen that $S_{\lceil \log_2 n \rceil}$ with this replacement is identically 1 (this can be proved, e.g., by induction; for S_1 it is trivially true, and the general statement follows from (3.3.7)).

Let there be given an arbitrary symmetric function g. It is defined by a subset $M \subseteq \{0,1,\ldots,n\}$ of possible values of N_1. Each branch of length $\lceil \log_2 n \rceil$ in $T(S_{\lceil \log_2 n \rceil})$ corresponds to one value of N_1 (given by the binary number $j(\lceil \log_2 n \rceil),\ldots,j(1)$ where $x_{\lceil \log_2 n \rceil, j(\lceil \log_2 n \rceil)},\ldots,x_{1,j(1)}$ define the branch in question). If we remove branches of $T(S_{\lceil \log_2 n \rceil})$ corresponding to $\bar{M}$, thus obtaining the formula S', and perform the substitution defined previously to obtain the formula p, we obtain a representation for g.

We have $O(L(S')) \leq n$ and thus $O(L(P)) \leq n^{4+\epsilon(n)}$ where $\epsilon \to 0$ as $n \to \infty$. Thus, if a basis Φ is given that contains all operators used to obtain P, then $L(g,\Phi) \leq n^{4+\epsilon(n)}$.

In Lu70 Lupanov announced a result of Khrapchenko to the effect that an arbitrary symmetric function is of length $\leq n^{4.93}$. Since the assumption were not made explicit, and the result itself is unavailable as of this writing, no exact comparison can be made with the estimate above.



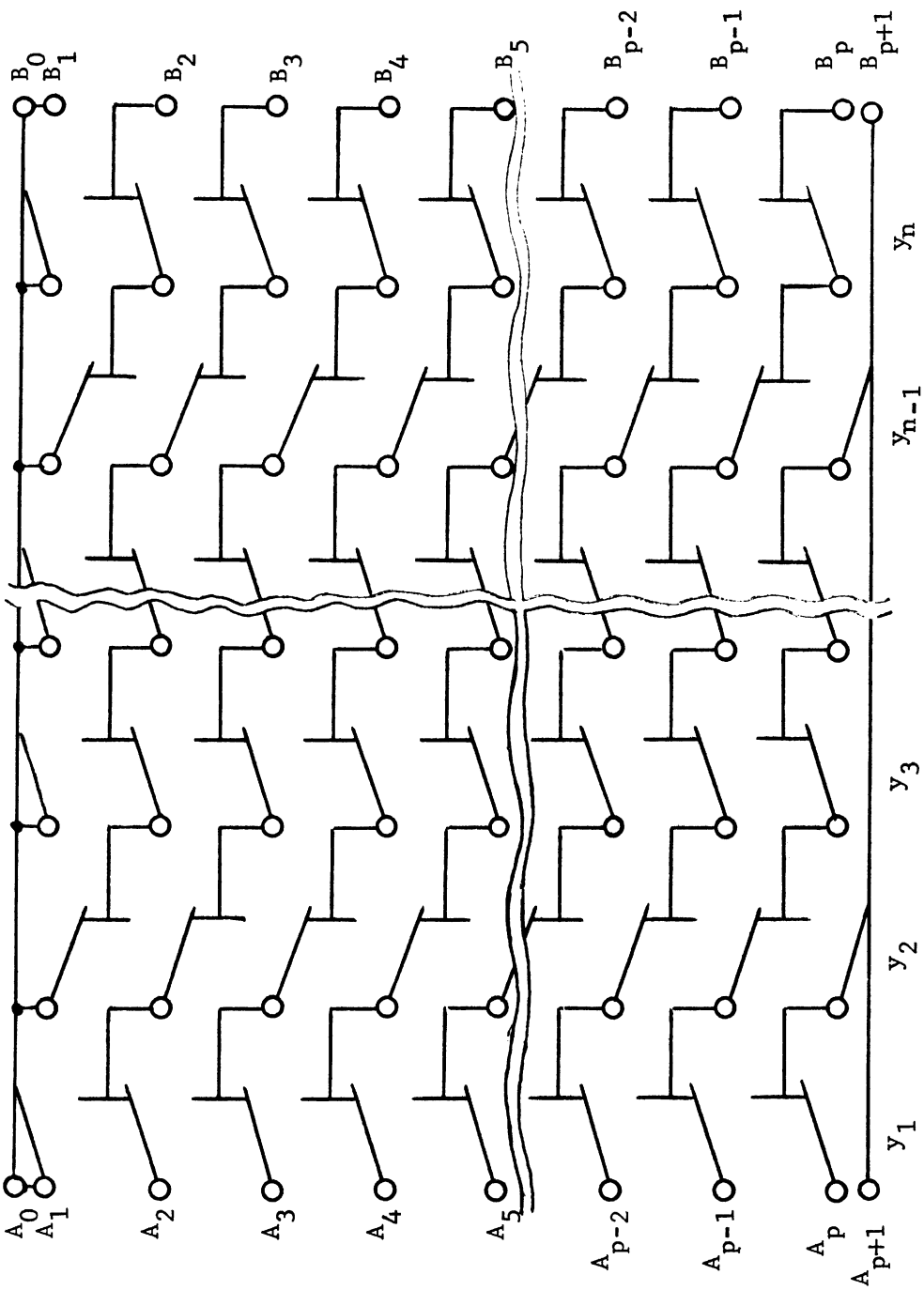
$\Gamma(\varphi_i)$ where φ_i has the I^0-property (only arrows labeled with 0 and 1 are drawn). Γ' denotes the set of nodes $\varphi_i(x_m x_{m-1} \cdots x_2 0, a_i)$ where m is as described in the text and $x_m, \ldots, x_2$ may assume arbitrary values in {0, 1}.

Fig. 3.1

```
0 0 0 0 0 0 0 0
0 0 0 0 1 1 1 1
0 0 0 0 1 1 0 1
0 0 0 0 1 0 0 1
0 1 1 0 1 0 1 1
0 1 1 1 1 0 1 1
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
```

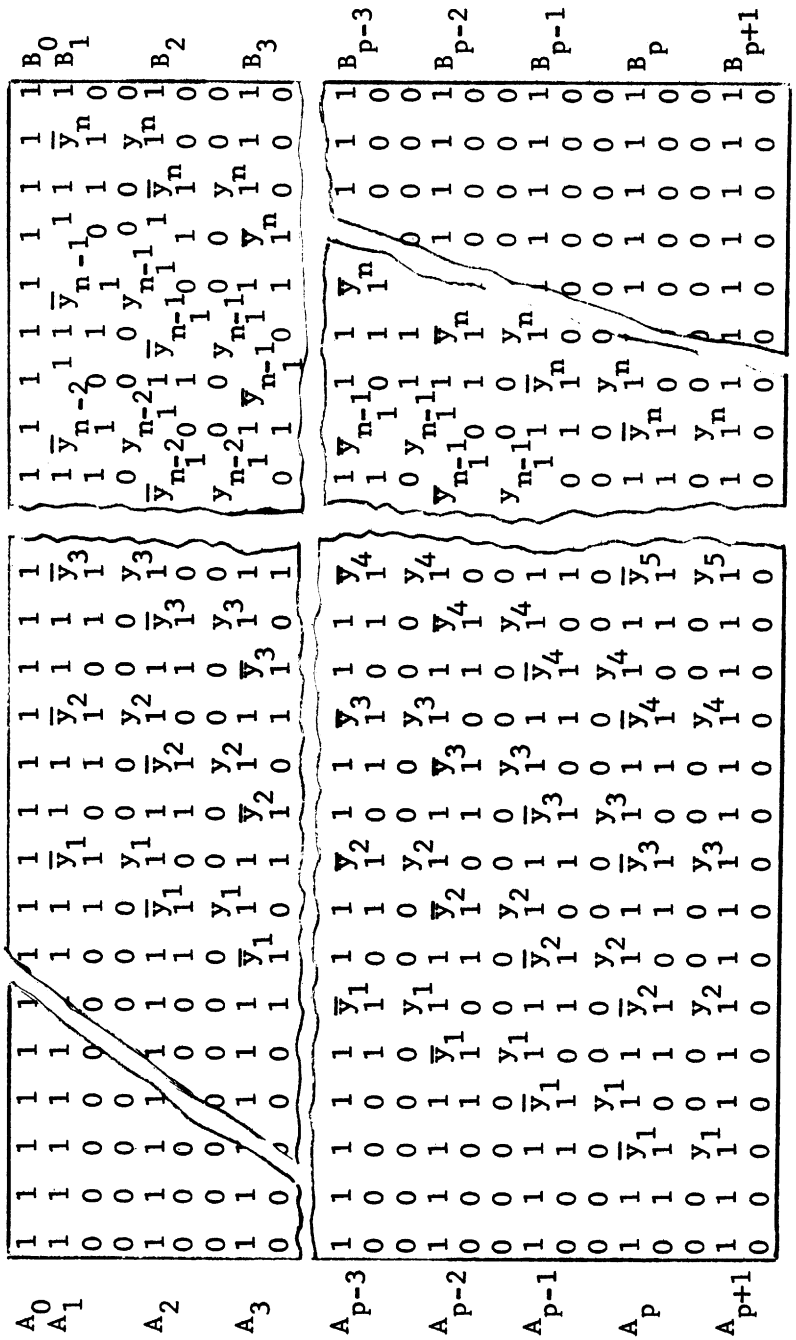
A connected pattern of 1's

Fig. 3.2



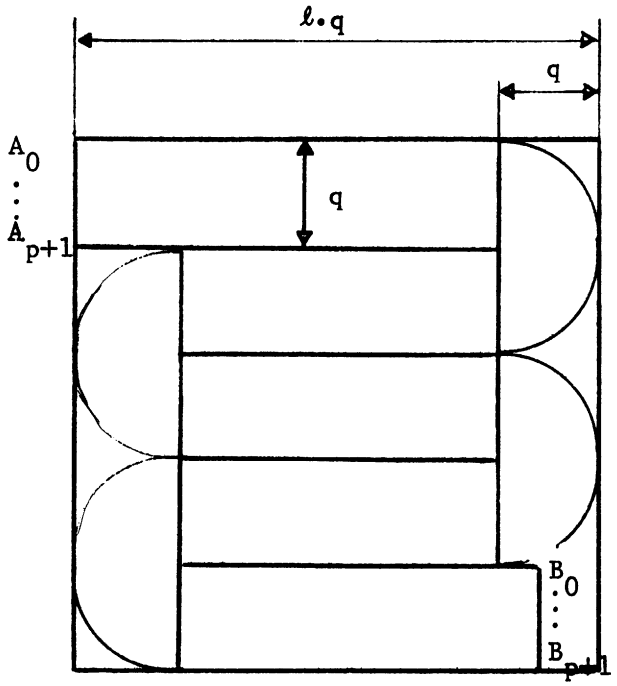
Contact network S_n^p for s_n^p

Fig. 3.3a



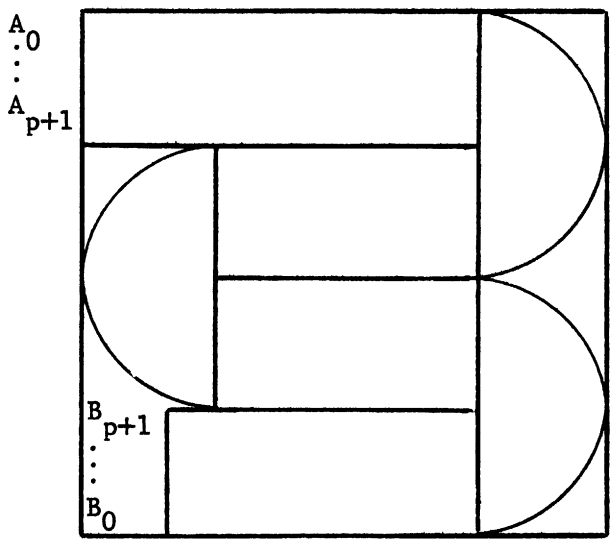
Planar connectivity simulation of S_n^p

Fig. 3.3b



T_n^p for $\ell = 5$

Fig. 3.4a



T_n^p for $\ell = 4$

Fig. 3.4b

CHAPTER FOUR

CYCLIC PERCEPTRONS

The perceptron has already been discussed in 1.6. In the beginning of this chapter, we will first expand on that discussion in order to further motivate the study of cyclic perceptrons.

The classical perceptron (for references on the subject see [Mi69] became the subject of extensive research centered around concepts such as pattern recognition, learning, adaptive behavior, etc. A whole myth had been created around it -- about its capabilities and its potential for use. The thing that attracted people most were its ability to learn from experience and its simplicity -- it combines many small decisions, the values of the functions φ_i, into a final decision by considering their weighted sum. Minsky and Paper deflated this myth by showing that such a scheme has its inherent drawbacks. In particular, it cannot compute predicates such as connectivity.

The most general intuitive basis for the result that the connectivity preciate cannot be represented by a perceptron is the following: First of all the reasoning makes sense only if the complexity of the functions φ_i is limited in some way; if not, we can choose φ_i to be the function that we desire to represent and then it can be represented by a perceptron trivially. Minsky and Papert use the order and diameter restrictions (see [Mi69]). The former is also used by us.

Suppose we want to represent connectivity. Then, if the φ_i's are bounded in complexity (so the reasoning goes), the weighted sum is too simple a function

to be able to integrate all the information that is required in computing connectivity.

We set out to apply the same basic reasoning to models where the integrating function is constructed out of finite operators. In particular, we choose addition in a finite field because of the unique representation property for functions in such a field (see 4.5) which makes proofs rather simple, and because of the purely formal resenblance to perceptrons. One particularly interesting aspect of using addition in a finite field as the integrating function is that one proof of the inability of perceptrons to compute connectivity is based on the reduction of connectivity to addition mod 2. However, this function is precisely the simplest one possible in GF(2). This underscores the need to make different reductions for different models of computation that are presumed to be incapable of computing connectivity.

In this chapter we shall limit ourselves to Boolean functions.

We introduce cyclic perceptrons formally:

4.1 Definition

$GF(p^k)$ is the finite field consisting of p^k elements. Φ (the basis) is an infinite set of Boolean functions $\{0,1\}^\omega \to \{0,1\}$ such that each $\varphi \in \Phi$ (ω is the first infinite ordinal) depends on a finite number of arguments. Elements of Φ are assumed to be ordered (in an arbitrary way). Then a (p,k)-perceptron (over Φ) is a pair $P = (a,Y)$, where $\underline{a}$ is an ω-vector such that the ith component $a_i \in GF(p^k)$ and $a_i \neq 0$ for only finitely many values of i; $Y \subseteq GF(p^k)$.

Given a function f: $\{0,1\}^{\omega} \rightarrow \{0,1\}$, we will denote the set of arguments on which it depends by S(f).

Let P = ($\underline{a}$,Y) be a (p,k)-perceptron. Then P will represent the predicate (Boolean function)

$$f = [\sum_{i=0}^{\infty} a_i \varphi_i \in Y] \tag{4.1}$$

where the value of $\varphi_i \in \{0,1\} \subseteq GF(p^k)$. Obviously, $S(f) \subseteq \bigcup_{i \in \{j:\ 0_j \neq 0\}} S(\varphi_i)$

We will indicate the function represented by a (p,k)-perceptron P as in (4.1), or simply [P].

Let us recall a concept from [Mi69]. Given a (p,k)-perceptron P = ($\underline{a}$,Y) over a certain basis Φ, its order (ord(P)) is $\max_{i \in \{j:\ a_j \neq 0\}} (S(\varphi_i))$.

We can also introduce the order of a function.

4.2 Definition

The (p,k)-order of a Boolean function f over a given basis Φ ((p,k)-$ord_{\Phi}(f)$) is the smallest ℓ such that there exists a (p,k)-perceptron of order ℓ representing f. If no such perceptron exists, the (p,k)-order of f is defined to be ∞.

Let Ω be the set of all Boolean functions with finite support. Note then that for an arbitrary Boolean function f, (p,k)-$ord_{\Omega}(f)$ is finite and $\leq S(f)$, for all primes p and arbitrary k. Also note that for an arbitrary basis Φ

$$(p,k)\text{-}ord_{\Omega}(f) \leq (p,k)\text{-}ord_{\Phi}(f) \tag{4.2}$$

We show now, as is done in [Mi69], that we can choose for the basis a more restricted set. Let the set of arguments of the basis functions be $\Xi = \{x_1, x_2, \ldots\}$; then we define the set of masks $M = \{ \bigwedge_{i \in S} x_i : S \text{ is a finite subset of } \mathbb{N} \}$. A convenient way of ordering M is to assign to $\varphi \in M$ the binary number $b_j b_{j-1} \ldots b_1$ where $b_k = 1$ if and only if x_k appears in the conjunction defining φ.

4.3 Proposition

Any Boolean function f can be represented by a (p,k)-perceptron over M for any prime p, and arbitrary k.

The proof is the same as that of Theorem 1.5.1 in [Mi69], i.e., we utilize the following correspondence between Boolean operations and operations in $GF(p^k)$ if the variables assume only the values 0 and 1:

$$x_1 \wedge x_2 \sim x_1 \cdot x_2,\ x_1 \vee x_2 \sim x_1 + x_2 - x_1 \cdot x_2,\ \bar{x} \sim 1 - x$$

If f is a function of n arguments, then from its disjunctive normal form, by using this correspondence and by multiplying out afterwards, we obtain the following representations for f:

$$\sum_{i=0}^{2^n-1} a_i x_1^{\sigma_{i1}} x_2^{\sigma_{i2}} \cdots x_m^{\sigma_{in}} = 1$$

where σ_{ij} is the jth bit of the binary representation of i and $a_i \in GF(p^k)$. Note that the mask φ_i (see the ordering above) is represented by the monomial with exponents corresponding to the binary representation of i.

Theorem 1.5.3 of [Mi69] also holds in our case. We state it as

4.4 Proposition

The following holds for an arbitrary Boolean function f, an arbitrary basis Φ, and an arbitrary integer k and prime p:

$$(p,k)\text{-ord}_M(f) \leq (p,k)\text{-ord}_{\Phi}(f)$$

Proof

The same as in [Mi69].

Note that if we take Ω for the basis in Proposition 4.4, and combine it with (4.2), we obtain that (p,k)-perceptrons over M achieve minimal order.

We state without proof the following well-known

4.5 Lemma

Every function $GF(p^k)^n \to GF(p^k)$ can be uniquely represented as a polynomial in n variables over $GF(p^k)$ that is at most of degree p^k-1 in each variable. (see, e.g., [La67].)

It has already been noted that we will be interested in whether a function can be represented by a (p,k)-perceptron with a limitation on its order. For this we need the following

4.6 Definition

A sequence of Boolean functions $f_1,f_2,\ldots$ of $1,2,\ldots$ arguments is of finite[†] (p,k)-order (over a given basis Φ) if there exists a finite r such

[†]Bounded would be a better word, but we conform to the terminology of [Mi69]

that for all i $(p,k)\text{-ord}_{\Phi}(f_i) \leq r$.

Let there be given a (p,k)-perceptron $(\underline{a},Y)$. If $Y = \{y_1,\ldots,y_m\}$, then, recalling that $GF(p^k)$ is a vector space of dimension k over GF(p), and designating the jth component of a_i by a_{ij} (similarly for $y_h \in Y$), we have

$$\left[\sum_{i=0}^{\infty} a_i \varphi_i \in Y\right] = \bigvee_{h=1}^{m} \bigwedge_{j=1}^{k} \left[\sum_{i=0}^{\infty} a_{ij} \varphi_i = y_{hj}\right] \tag{4.3}$$

We can restrict the diversity of perceptrons we are dealing with by noting

4.7 Proposition

Let Φ be a basis closed under conjunction (i.e., $\varphi, \psi \in \Phi \Rightarrow \varphi \wedge \psi \in \Phi$). If a Boolean function f is of finite (p,k)-order over Φ, then it is of finite (p,1)-order (but the order may change).

Proof

We have $f = [(\underline{a},Y)]$ where $(\underline{a},Y)$ is a (p,k)-perceptron. Suppose $|Y| = m$ and the (p,k)-order of f is ℓ. From (4.3) we have

$$f = \bigvee_{h=1}^{m} \bigwedge_{j=1}^{k} \left[\sum_{i=0}^{\infty} a_{ij} \cdot \varphi_i = y_{hj}\right] \tag{4.4}$$

where $a_{ij}, y_{hj} \in GF(p)$.

By Lemma 4.5, we know that for all $a \in GF(p)$ there always exists a polynomial $P_a(x)$ over GF(p) of degree p-1 which takes on the value of 1 if x = a and is 0 otherwise (the degree follows from the number of zeros of the polynomial). Thus substituting the Boolean operations with the field operations introduced in the proof of Proposition 4.3, we obtain from (4.4)

$$f = Q\left(\prod_{j=1}^{k} P_{y_{ij}}\left(\sum_{i=0}^{\infty} a_{ij}\cdot\varphi_i\right),\ldots,\prod_{j=1}^{k} P_{y_{mj}}\left(\sum_{i=0}^{\infty} a_{ij}\cdot\varphi_i\right)\right) \qquad (4.5)$$

where $Q(x_1,\ldots,x_m)$ is the polynomial (of degree m) that represents the Boolean function $\bigvee_{h=1}^{m} x_h$[†]. Each $P_{y_{ij}}$ is of degree p-1. Hence f can be expressed as a polynomial in the φ_i's of degree $\leq m\cdot(p-1)$. Obviously, φ_i^j for $j > 1$ can be replaced by φ_i since it assumes only the values 0 and 1. Also, $\varphi\cdot\psi$ represents the function $\varphi\wedge\psi$ and $|S(\varphi \wedge \psi)| \leq |S(\varphi)| + |S(\psi)|$; thus, if the basis is closed under conjunction (as, e.g., Ω or M), (4.5) describes a (p,1)-perceptron for f of order $\leq m\cdot(p-1)\cdot\ell$. □

Remark

Incidentally, this proof also shows that we can assume the cardinality of Y to be 1.

Since we shall subsequently be concerned only in whether the order of certain functions is finite or not, we will be able to limit ourselves to (p,1)-perceptrons. For convenience, we will write simply "p-perceptrons". Also, we will be only concerned in whether there exists a basis over which a function is of finite order. This is equivalent to whether a function is of finite order over M.

[†] $Q(x_1,\ldots,x_m)$ is obtained by using $y_1 \vee y_2 \sim y_1 + y_2 - y_1\cdot y_2$ recursively; i.e., $Q(x_1,\ldots,x_m) = Q(x_1,\ldots,x_{m-1}) + x_m + x_m\cdot Q(x_1,\ldots,x_{m-1})$. If Q is a polynomial over GF(2), then $Q = \bigoplus \prod_{y\in S} y$ where the sum ranges over all nonempty subsets $S \subseteq \{x_1,\ldots,x_m\}$.

We first turn our attention to the case when p = 2. Instead of "2-perceptron", we will say "Boolean perceptron".

From Lemma 4.5 we conclude that every Boolean function can be uniquely represented as a polynomial over GF(2) that is at most of degree one in each variable (a Boolean polynomial).

Noting that the terms of a Boolean polynomial represent marks, we conclude that every Boolean function f has a unique representation as a Boolean perceptron over M. Furthermore, by Proposition 4.4, this representation is a minimal order representation for f. Note then that $2\text{-ord}_M(f)$ corresponds to the degree of the Boolean polynomial for f.

This unique representation property allows us to establish the minimal order of certain interesting predicates very easily. As in 3.2, we are again interested only in functions $\{0,1\}^{n^2} \rightarrow \{0,1\}$ that are interpreted as functions of nxn patterns of 0's and 1's. In particular, we are interested in the Boolean function of n^2 variables c_n (introduced in 3.2) and $e_{n,k}$ (the Euler number of a pattern of 1's on a square array of 0's and 1's is equal to k). It is well known (see, for example [Mi69]) that the Euler number of a planar figure is the difference between the number of its components and the number of its holes. If we use the notion of connectivity introduced in 3.2, then the Euler number of the pattern in Fig. 4.1 is 1.

4.8 Theorem

The connectivity predicate is not of finite 2-order over M (hence, over any basis).

Proof

We use the One-in-a-box construction introduced in [Mi69]. Before proceeding, however, we must define certain auxiliary predicates. n, the size of the pattern is assume odd (henceforth, we will suppress the subscript n in the notation for functions). The variables representing positions in the square array will, as usual, be denoted by x_{ij} for $1 \leq i, j \leq n$. Then we define

$$\begin{aligned} r = &(x_{11} \wedge x_{12} \wedge \ldots \wedge x_{1n}) \wedge \\ &(x_{31} \wedge x_{32} \wedge \ldots \wedge x_{3n}) \wedge \ldots \wedge \\ &(x_{n1} \wedge x_{n2} \wedge \ldots \wedge x_{nn}) \end{aligned}$$

and

$$\begin{aligned} s = &(x_{21} \vee x_{22} \vee \ldots \vee x_{2n}) \wedge \\ &(x_{41} \vee x_{42} \vee \ldots \vee x_{4n}) \wedge \ldots \wedge \\ &(x_{n-1,1} \vee x_{n-1,2} \vee \ldots \vee x_{n-1,n}); \end{aligned}$$

i.e., r is 1 only on patterns with odd rows consisting exclusively of 1's, and s is 1 only on patterns where each even row has at least one 1 (the One-in-a-box predicate). Then,

$$r \wedge c = r \wedge s \tag{4.6}$$

(c is the connectivity predicate).

Now, for arbitrary functions f, g, h, if $h = f \wedge g$, then $2\text{-ord}_M(h) \leq 2\text{-ord}_M(f) + 2\text{-ord}_M(g)$; i.e.,

$$2\text{-ord}_M(g) \geq 2\text{-ord}_M(h) - 2\text{-ord}_M(f) \tag{4.7}$$

Replacing h by $r \wedge s$, f by r, and g by c we obtain

$$2\text{-ord}_M(c) \geq 2\text{-ord}_M(r \wedge s) - 2\text{-ord}_M(r) \qquad (4.8)$$

We have $2\text{-ord}_M(r) = \frac{n+1}{2} \cdot n$; $2\text{-ord}_M(h) = \frac{n-3}{2} \cdot n$ (recall the Boolean polynomial for $\bigvee_{i=1}^{m} x_i$ described in the footnote on p. 80) $2\text{-ord}_M(r \wedge s) = n(n-1)$ (because ;the Boolean polynomial representations of r and s have no variables in common). Using this we obtain from (4.8) $2\text{-ord}_M(c) \geq \frac{n(n-3)}{2}$; i.e., the 2-order over M of the connectivity predicate is not finite. □

We next establish

4.9 Theorem

The predicate "the Euler number of a pattern equals k" is not of finite 2-order.

Proof

We again consider the case when M is the basis. The general case follows from Proposition 4.4. n is the size of the pattern. We need to consider a subset $T \subseteq S = \{x_{ij} : i+j \text{ even}\}$ (note that all points of S are disconnected from each other, in the sense we use this word). $|T| = t$ will be determined subsequently.

We define the following predicates

$pr = 1$ if and only if all points of $\bar{T}$ are 0; i.e.,

$$pr = \prod_{x \notin T} (1 \oplus x)$$

$q = 1$ if and only if k points of T are 1; i.e.,

$$q = \bigoplus \prod_{x\in U} x \cdot \prod_{y\in T-U} (1 \oplus y)$$

where the sum ranges over all possible subsets $U \subseteq T$ with $|U| = k$. When the expression for q is multiplied out each term produces exactly one term of the form $\prod_{x\in T} x$ and thus the above Boolean polynomial is of degree t if and only if the number of terms in the sum is odd. The number of terms is $\binom{t}{k}$. But $\binom{2^{\ell}-1}{k}$ is odd for all $0 \le k \le 2^{\ell}-1$ and all ℓ.† Thus if $t = 2^{\ell}-1$, $0 \le k \le t$ then 2-ord(q) = t. Also, 2-ord(pr) = $|\bar{T}| = n^2-t$.

Recalling once again the e_k is the difference between the number of components of a figure and the number of holes, we have the relationship

$$pr \wedge e_k = pr \wedge q$$

Again using (4.7) with $g = e_k$, $h = pr \wedge q$, $f = pr$ we obtain

$$2\text{-ord}_M(e_k) \ge n^2 - |\bar{T}| = 2^{\ell}-1$$

No matter how large we choose ℓ, we can find an n such that we can obtain a set T with $|T| = 2^{\ell}-1$. Thus, the Euler predicate is not of finite order over M. □

Theorems 4.8 and 4.9 can be extended to p-perceptrons for arbitrary p. The generalization will only be indicated for Theorem 4.8.

---

†Proof: First show that $\binom{2^{\ell}}{k}$ is even for all ℓ and all $k \neq 0, 2^m$. This is done by induction. Now observe that due to $\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$, and the fact that $\binom{2^{\ell}-1}{1}$ is odd, $\binom{2^{\ell}-1}{2}$ is also odd (for otherwise $\binom{2^{\ell}}{2}$ would not be even). We can continue this way and establish the claim.

The obvious difficulty is that Boolean functions do not have a unique representation as polynomials over GF(p) for $p > 2$. Specifically, in the case of Boolean perceptrons over M, we were able to reduce the problem of the order of connectivity to the order of r and s (see above). The orders of these predicates (equal to the degrees of the corresponding Boolean polynomials) were easily computed due to Lemma 4.5.

Suppose c is of finite p-order over M (we have already remarked that this brings no loss of generality) for some $p > 2$. Due to Proposition 4.7 we can assume that we have an expression of the form of (4.5) for c_n. When multiplied out, we obtain

$$c_n = \sum_{i=0}^{\infty} a_i m_i \mod p \qquad (4.9)$$

where m_i is the monomial representing the ith mask. m_i is of degree one in each variable, and the values of the variables are restricted to {0,1}. Since the perceptron from which we obtained (4.9) is finite order, we can assume that the degree of (4.9) is $\leq \ell$.

We can now extend c_n to the domain GF)p) (i.e., to square patterns $A = \{b_{ij}\}$, $1 \leq i, j \leq n$, and $b_{ij} \in GF(p)$) by defining $c_n'(A) = 1$ if and only if $c_n(f(A)) = 1$ where $f\{b_{ij}\}) = \{c_{ij}\}$ such that $c_{ij} = 1$ if $b_{ij} = 1$, otherwise $c_{ij} = 0$. We have from (4.9)

$$c_n' = \sum_{i=0}^{\infty} a_i . m_i' \mod p \qquad (4.10)$$

where m_i' is obtained from m_i by replacing each variable x_j by $P_1(x_j)$ such that $P_1(x_j) = 1$, otherwise $P_1(x_j) = 0$ (see the proof of Proposition 4.7 how to obtain $P_1(x_j)$). Now we have a total function c_n' over GF(p), and thus its

polynomial representation obtained by multiplying out (4.10) must be unique. Since the degree of (4.9) is $\leq \ell$,

$$\text{the degree of the polynomial (4.10)} \leq (p-1)\ell \tag{4.11}$$

Another estimate of the degree of the polynomial for c_n' is obtained using the predicates r' and s' obtained from r and s of Theorem 4.8 similarly as c_n' was obtained from c_n. The polynomial P_r, representing r', is obtained from the polynomial representing r by substituting each variable x with $P_1(x)$. Similarly, for the polynomial representation P_s, of s'. The degrees of $P_{r'}$ and $P_{s'}$ are then found to be $\leq \frac{n+1}{2}\, n\cdot(p-1)$ and $\leq \frac{n-3}{2}\, n(p-1)$ respectively ($2\text{-ord}_M(r)$ and $2\text{-ord}_M(s)$ multiplied by $\deg(P_1)$). Since the analogs of (4.6) and (4.8) again hold, we obtain that $\deg(P_{c'})$ is not bounded, contradicting (4.11), and thus also the finite order of c.

Note that the obvious generalization, i.e., if a function is of finite 2-order, then it is also of finite p-order (over M), is not true: Consider the Boolean function $\bigoplus_{i=1}^{n} x_i$. We will investigate the degree of the polynomial representation of $\oplus'$ (obtained in the same way from $\oplus$ as c' was from c above). If a p-perceptron over M of finite order exists for $\oplus$, then we obtain a polynomial representation of bounded degree for $\oplus'$ similarly as (4.10) was obtained from (4.9). On the other hand we have the following representation for $\oplus$

$$\sum \prod_{i\in S} x_i \prod_{i\notin S} (1-x_j) \bmod p \tag{4.12}$$

where S ranges over all subsets of $\{1,\ldots,n\}$ of even size. When multiplied out, each term produces exactly one monomial of the form $\prod_{i=1}^{n} x_i$. The number

of such monomials appearing in the developed form of (4.12) is

$$\sum_{i=0}^{\lfloor \frac{n}{2} \rfloor} \binom{n}{2i} \equiv 2^{n-1} \pmod{p}$$

(use the identity $\binom{n}{i} = \binom{n-1}{i} + \binom{n-1}{i-1}$). Since this number is not 0 mod p, (4.12) yields a p-perceptron over M of order n for Φ. If x_i is replaced by $P_1(x_i)$ we obtain a (unique) polynomial representation for Φ' of degree $n \cdot (p-1)$, contradicting the existence of a finite order p-perceptron over M for Φ.

of such monomials appearing in the developed form of (4.12) is

$$\sum_{i=0}^{\lfloor \frac{n}{2} \rfloor} \binom{n}{2i} \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

(use the identity $\binom{n}{i} = \binom{n-1}{i} + \binom{n-1}{i-1}$). Since this number is not 0 mod p, (4.12) yields a [illegible] of order n for Θ. If x_i is replaced by $P_i(x_i)$ we obtain a (unique) polynomial representation for Θ' of degree $n \cdot (p-1)$, contradicting the existence of a finite order p-perceptron over M for Θ.

CHAPTER FIVE

PATTERN COUNTING MACHINES

In this chapter we shall permit ourselves a certain degree of informality.

We are again concerned with the power of machines that combine a large number of "local" computations through an integrating function. Only this time we shall not be limited to functions that can be represented as a combination of finite operators.

This class of machines again operates on square patterns of 0's and 1's. The operation is divided into two phases: In Phase I the pattern is scanned with a square "window" of a certain size. Each time a nonzero pattern appears in the window, we take note of it (there is a finite number of nonzero patterns since the window is of finite size). At the end of the scan we have a count of the various patterns, and we are then allowed to utilize this data in Phase II which consists of computing the value of a partial recursive function for this data. The formalization of this model is obvious and we omit it.

What can such a machine do? Clearly the computation of this machine is divided into a local phase and a global phase, so that it fits into the broad class of problems considered in [Mi69] and Chapter Four.

Note that the boundedness of the window size is essential. If we insisted only that the window contain a given number of points, but otherwise allowed it to be of any shape with arbitrary distances between its points, then Phase II could reconstruct the whole figure as was observed already in [Mi69].

We again inquire whether these machines can recognize the familiar

topological predicates connectivity and Euler number.

5.1 Theorem

Pattern Counting Machines (PCM) cannot recognize the connectivity predicate.

Proof

We need only exhibit two patterns, one connected and the other disconnected, with the same pattern spectrum. In this case, no algorithm of Phase II could establish the difference between them.

Two such patterns are given in Fig. 5.1.

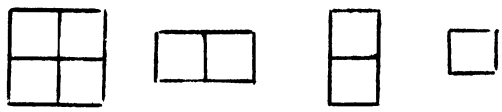
Specifically, these patterns are equivalent under windows of size 2x2. However, it is easily seen that increasing the dimensions of the patterns in Fig. 5.1 linearly by a factor of k makes them equivalent under windows of size up to k + 1. We can arrive at this conclusion by setting up a 1-1 map between occurrences of the same pattern in the window in the two patterns

5.2 Theorem

PCM's can compute the Euler number.

Proof

It is shown in [Mi69] how to compute the Euler number from the spectrum of patterns of the shape



□

Before proceeding, we need a notion of continuous deformation. Pattern B can be obtained from pattern A by continuous deformation if B arises from A by a sequence of additions or deletions of 1's of the following kind: Let us fix attention on a 3x3 square with the central position in the place of the 1 being added (deleted). For simplicity assume that the boundary positions are always 0. Each position of the periphery of the square which has a 1 in it is either connected or disconnected to another 1 on the periphery (not necessarily by a path in the square). This set of connections may be described by a symmetric 8x8 0-1 valued connection matrix, i.e., $a_{ij} = 1$ if and only if the ith and j^{th} positions on the periphery have 1's and are connected. The proposed addition (deletion) is permitted only if (1) the connection matrix remains unchanged as a result of it, and (2) there is a 1 adjacent to the proposed addition (deletion).

Any predicate whose value remains unchanged if the pattern A is replaced by B, obtained from A by continuous deformation, is called a topological predicate. We assert without proof that connectivity and Euler number are topological predicates. The reader is warned, however, that there is a pitfall in proving this fact for the Euler number predicate. The number of holes in Fig. 5.2 should be one, not two (i.e., 0's are connected diagonally in addition to their usual connections). This is discussed more fully in [My71]. However, if the holes are sufficiently large (so that all the 0's in them are connected in the usual way) this difficulty is not encountered.

5.3 Theorem

Any topological predicate recognized by a PCM must be a function of the Euler number.

Proof

We will have established the theorem if we succeed in showing that, given any PCM P computing a topological predicate, then for two figures X_1 and X_2 with EULER(X_1) = EULER(X_2), we also have $P(X_1) = P(X_2)$.

In [Mi69] it is shown that for every figure X there exists an "Euler canonical" figure C(X) such that EULER(X) = EULER(C(X)); and if for two figures X_1, X_2, EULER(X_1) = EULER(X_2), then $C(X_1) = C(X_2)$. If the Euler number of X is $n > 0$, then C(X) consists of n components without holes. If the Euler number of X is $n \leq 0$, then C(X) consists of 1 component with $-n+1$ holes.

We will show that we can deform any figure X into **C**(X) without changing the value of P.

The deformations available to us are:

(1) Continuous deformation. If we subject X to this kind of deformation, then P(X) remains unchanged because it computes a topological predicate.

(2) Deformations that leave the pattern spectrum unaltered. By definition of PCM's.

As a consequence, we have

(3) Removal of components inside holes. To accomplish this without changing the value of P(X), we first apply (1) until the window cannot scan simultaneously an interior component and the wall of the hole in which it resides. Then it is obvious that the pattern spectrum will remain unchanged if we remove the component from inside the hole. After this we can apply (1) in the reverse direction.

If we are given two figures X_1 and X_2 with same pattern spectra, then we can add equally shaped holes to the figures in such a way that the pattern spectra remain the same. The holes have only to be placed in such a way that the window cannot scan any other boundary while scanning the newly introduced hole. We can then repeat this to add any number of holes.

Specifically, given two figures of the shape of A and B in Fig. 5.1, we can add holes in this way and still have the same pattern spectra. For example, C and D in Fig. 5.3 have the same pattern spectra for a sufficiently small window size. Note that given any two components, one of which has a hole, we may apply deformation (1) to obtain a figure proportional in dimensions to C and they apply (2) to obtain D. We call this sequence of deformations "cancelling a hole and a component".

We deform X into C(X) by cancelling as many holes and components as possible. We first apply (3) until no component remains within a hole. Then we may either have a hole and a component not containing this hole, or not. In the latter case, we are done. In the former, we select a hole and a component not containing it and cancel them. After this we are left with one less component and hole. We repeat this until we arrive at C(X).

We can summarize the results on the recognition of topological predicates contained in [Mi69], Chapter Four, and in the present section in the following table

Recognition of Predicates	Cyclic Perceptrons	PCM	Classical Perceptrons
Connectivity	NO	NO	NO
Euler	NO	YES	YES
Other Topological Predicates	?	Functions of Euler number only	Functions of Euler number only

Thus, all these results support the conjecture expressed in [Mi69] that no "local-global" computer can recognize connectivity.

It appears, however, that all models are extremely sensitive to alterations. We have already mentioned how PCM's can be converted into universal machines with the removal of the restriction on the size of the window. A. R. Meyer noticed that ordinary perceptrons may be modified to recognize any Boolean function with order one. Instead of $\sum_{i=0}^{\infty} \alpha_i \varphi_i \geq 0$ consider $\sum_{i=0}^{\infty} \alpha_i \varphi_i \in Y$ for some subset of integers Y. Now we can choose the coefficients a_i in such a way that the sums of the coefficients in no two subsets of the set K of all coefficients is the same, ie.,

$$\forall(X,Y \subseteq K)\ [\ \sum_{a_i \in X} a_i = \sum_{a_i \in Y} a_i \Rightarrow X = Y]$$

We can define the coefficients inductively, i.e., choose a_n to be greater than $\sum_{i=1}^{n-1} a_1$. This is in the spirit of stratification (see [Mi69]). Now notice that if Φ is the set of masks of order 1, then a Boolean function φ

```
0 0 0 0 0 0 0        0 0 0 0 0 0 0
0 0 0 1 0 0 0        0 0 0 0 0 0 0
0 0 0 1 0 0 0        0 0 1 1 1 0 0
0 0 0 1 0 0 0        0 0 1 1 1 0 0
0 0 0 0 0 0 0        0 0 0 0 0 0 0
```

B A

Two figures with the same 2x2 pattern spectra

Fig. 5.1

```
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 1 1 1 1 0 0 0
0 0 1 1 0 1 0 0 0
0 0 1 0 1 1 0 0 0
0 0 1 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
```

The number of holes in this pattern is one

Fig. 5.2

D C

Cancelling a hole and a component

Fig. 5.3

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	0
0	1	0	1	0	1	1	0
0	0	0	1	1	1	1	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

A

0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	0	1	0	0	0	0
0	1	1	1	1	1	1	0
0	0	0	1	0	0	0	0
0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0

B

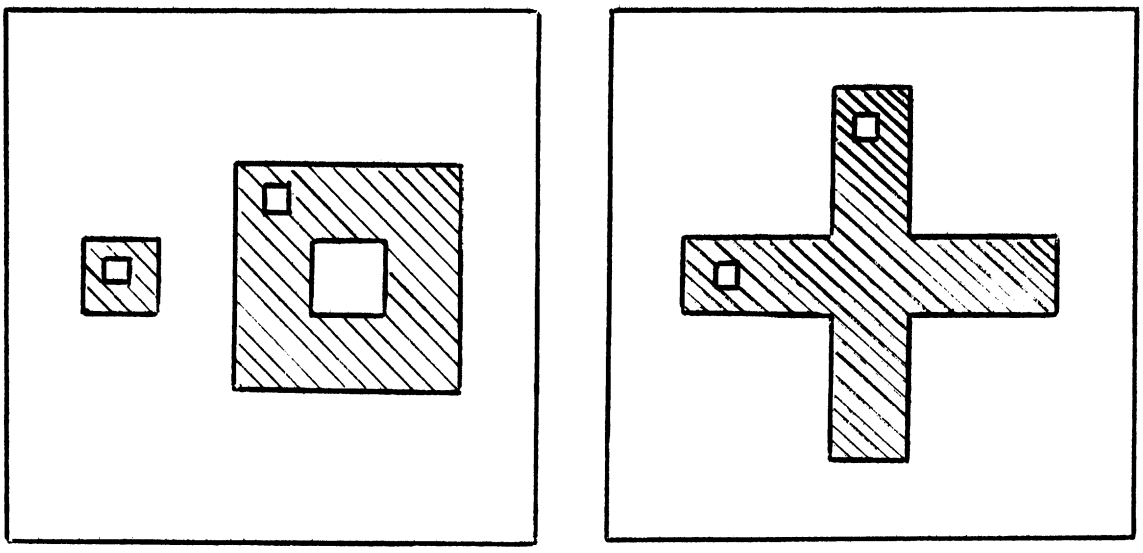
Two figures with the same 2x2 pattern spectra

Fig. 5.1

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	0	0
0	0	1	1	0	1	1	0	0
0	0	1	0	1	1	1	0	0
0	0	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

The number of holes in this pattern is one

Fig. 5.2



C D

Cancelling a hole and a component

Fig. 5.3

APPENDIX A

CERTAIN PROPERTIES OF SHORT FORMULAS

The purpose of this appendix is to modify certain results of [Ho68] in the light of our different requirements. Our goal is Lemma A.9 which is used directly in the proof of Theorem 2.2.2. We prove it by way of a series of intermediate results, none of which are used elsewhere.

In what follows we would frequently use the phrase "F is a formula in n variables over Φ, and such that no variable occurs more than k times". This will be abbreviated to "F is a (Φ,n,k)-formula". If any of the parameters is not present, we will replace it by *. For example, "F is a $(\Phi,*,k)$-formula" and "F is a $(\Phi,n,*)$-formula" mean "F is a formula over Φ, and such that no variable appears more than k times" and "F is a formula in n variables over Φ" respectively.

A.1 Definition

Let there be given the sequence of formulas $\underline{G} = (G_1(X_1,z),\ldots,G_{p-1}(X_{p-1},z),G_p(X_p))$. If $1 \leq i \leq p-1$, then G_i contains the distinguished variable z, occurring only once. X_i for $1 \leq i \leq p$ is nonempty and is either a singleton or $\subseteq \bigcup_{j<i} X_j$. Let F be an arbitrary formula, and $G = G_1(X_1,G_2(X_2,\ldots,G_{p-1}(X_{p-1}G_p(X_p))\ldots))$. If $F \equiv G$, then $\underline{G}$ is a nesting sequence of length p for F. If, in addition, the total number of occurrences of any variable (except z) in $\underline{G}$ is $\leq$ the corresponding number in F, then $\underline{G}$ is a proper nesting sequence for F.

A.2 Remark

Let $\underline{G}$ and G be as described in Definition A.1. Furthermore, let all X_i for $1 \leq i \leq p$ be singletons and distinct. Then G is equivalent to an e_{p-1}-component. Also, suppose X_i is arbitrary and G_i is a formula over Φ. Now replace all variables except possibly one in G_i for $1 \leq i \leq p-1$ by the constant a. Let the set of variables that have not been touched be Y. Then G_a^Y is equivalent to an e-component over Φ^a.

Let F be an arbitrary formula over Φ, $X \subsetneq S(F)$, and $a \in D$; then we would like to obtain a formula over Φ^a with the following properties: (1) $G \equiv F_a^X$ (2) $S(G) = X$, and (3) the number of occurrences of any variable of X in G is $\leq$ the corresponding number in F. G can be obtained by a straightforward replacement of operators in F such that the variable symbols that are replaced with a in forming F_a^X (and subformulas of F where S(F) consists entirely of such variable symbols) are removed, and the remaining operators are changed to preserve equivalence with F_a^X. More precisely, if $\varphi(F_1,\ldots,F_k)$ is a subformula of F, then if $S(F_i) \not\subset \bar{X}$ for all i, φ remains the same; if $S(F_i) \subset \bar{X}$ and $S(F_j) \not\subset \bar{X}$ for $j \neq i$, then φ is replaced with $\varphi(x_1,\ldots,x_{i-1},F_i,X_{i+1},\ldots,x_k)$ where all variables of F_i have been replaced with a (if there are more such indices i, we proceed in the obvious way); and if $S(F_i) \subset \bar{X}$ for all i, then φ is eliminated. This transformation will be called normalization and G will be denoted by $\text{norm}(F_a^X)$.

A.3 Lemma

If F is a $(\Phi,n,*)$-formula, then for any $p, q \geq 1$ and $a \in D$, if $n \geq \eta_6(p,q)$, there exists a subset $X \subseteq S(F)$ such that either

(1) $|X| = q$ and F_a^X is equivalent to a PC of the formulas $F_1,\ldots,F_r$ where $r \leq n_{max}$ and F_i for $1 \leq i \leq r$ is a formula over Φ^a such that each element of X occurs in at least two among $F_1,\ldots,F_r$, and the total number of occurrences of any $x \in X$ in $F_1,\ldots,F_r$ is $\leq$ the number of occurrences of x in F; or

(2) $|X|$ is arbitrary and F_a^X has a proper nesting sequence $\underline{G} = (G_1,\ldots, G_{p-1}, G_p)$ where G_i for $1 \leq i \leq p$ is a formula over Φ^a.

Proof

Assume there is no $X \subseteq S(F)$ such that F_a^X is as described in (1) of the statement of the lemma.

We will describe a (proper) nesting sequence extraction procedure (NSE) whose inputs will be a formula H over Φ^a and a set of variables Y. The output of NSE will be two formulas $H'(Z,z)$ and H''' over Φ^a such that Z is either a singleton or $\subseteq Y$; furthermore, $H_a^U \equiv H'(Z,H''')$ for some $U \subseteq S(H)$.

$\underline{G}$ will be obtained by the repeated use of NSE. Initially, the input of NSE will be $F = F_0$ and ϕ. In the first application of NSE, the output will be G_1 and F_1 (F_1 is an intermediate formula whose significance will be describe immediately). In general, the ith application of NSE will receive the input F_{i-1} and $\bigcup_{j \leq i} X_j$ and yield as output G_i and F_i. We will

show that if $n \geq \eta_6(p,q)$, we can apply NSE p-1 times and end up with F_{p-2} from which G_p is obtained as will be described below.

Description of NSE. The input to NSE is as describe above. Then we can distinquish two cases:

Case I. $L(H) = 1$. In this case we cannot apply NSE, and the output is undefined.

Case II. $L(H) > 1$. In this case we can assume that H has no unary operators; for suppose there exists a subformula J of H such that $J = \varphi(\psi(J_1,\ldots,J_r))$ where J_i for $1 \leq i \leq r$ is either a variable symbol or another subformula of H. In this case $\varphi\cdot\psi = \rho \in \Phi^a$ and we can replace J by the equivalent formula $\rho(J_1,\ldots,J_r)$. Similarly, if $J = \varphi(J_1,\ldots,\psi(J_i),\ldots,J_k)$, we can eliminate ψ because $\varphi(x_1,\ldots,\psi(x_i),\ldots,x_k) = \rho(x_1,\ldots,x_i,\ldots,x_k)$ $\in \Phi^a$ (thus, if a unary operator of H corresponds to an internal node of T(H), we can eliminate it by either of these two means; on the other hand, if a unary operator of H corresponds either to the root node or to a node next to a terminal node of T(H), then we can use only one of the two methods described). Now choose i' such that $S(H_{.i'})$ is maximal among $S(H_{.i})$ for $1 \leq i \leq r$. Since support is defined only for formulas, $S(H_{.i'})$ may be undefined if all arguments of the outermost operator of H are variable symbols. In this case replace one of them by the identity operator which is possible since $id \in \Phi^a$. Consider $H/H_{.i'} = K(Z,z)$. Again two cases can arise:

Case IIa. $Y \cap Z = \phi$. Choose any variable of Z, e.g., x, and let $H' = \text{norm}(K_a^{\{x,z\}})$. z is a distinquished argument (hence x is free).

In this case set $V = S(H_{.i})\text{-}Z \cup \{x\}$. The significance of V will be seen immediately.

Case IIb. $Y \cap Z \neq \phi$. $H' = \text{norm}(K_a^{Y \cup \{z\}})]$ z is again a distinguished argument $V = S(H_{.i'})\text{-}Z \cup Y$.

In both cases $H'' = \text{norm}((H_{.i'})_a^V)$.

Analysis of NSE. Let $|S(H)| = m$, and let us estimate $|S(H'')|$. Obviously, $|S(H_{.i'})| \geq \frac{m}{n_{max}}$. In the case that H results from a chain of applications of NSE to a formula F, and F does not satisfy (1) of the statement of the lemma, then we claim that in Cases IIa and b less than q variables are set to a in $H_{.i'}$. Suppose this is not true. Let the set of variables that is set to a on this occasion be W. Then $W \subseteq Z\text{-}\{x\}$ (Case IIa), or $W \subseteq Z\text{-}Y$ (Case IIb). In any case consider F_a^W. This is equivalent to $\varphi(H_{.i(1)})_a^W, \ldots, (H_{.i(s)})_a^W$ where $i(1), \ldots, i(s)$ are the indices corresponding to the subformulas $H_{.j}$ where all variables have not been replaced by a (if in $H_{.k}$ all variables have been replaced by a, then it is absorbed into φ). But then $\varphi(\text{norm}((H_{.i(1)})_a^W), \ldots, \text{norm}((H_{.i(s)})_a^W))$ satisfies (1) of the statement of the lemma. A contradiction.

Thus, $|S(H'')| \geq \frac{m}{n_{max}} - q + 1$

Hence, if we define

$$\eta_6(1,q) = 1$$

$$\eta_6(p+1,q) = (\eta_6(p,q)+q-1)\cdot n_{max},$$

i.e.,

$$\eta_6(p,q) = n_{max}^{p-1}\cdot q + \frac{n_{max}^{p-1}-1}{n_{max}-1}\cdot(q-1)$$

for $p,q \geq 1$ (and if F does not satisfy (1)), we will be able to apply NSE p-1 times and obtain F_{p-1}. G_p can then be obtained as follows: If $S(F_{p-1}) \cap S(G_i) = \phi$ for $1 \leq i \leq p-1$, then choose any variable $y \in S(F_{p-1})$ and obtain G_p from $(F_{p-1})_a^{\{y\}}$ by normalization; otherwise, denoting $\bigcup_{i=1}^{p-1} S(G_i)$ by U, obtain F_{p-1} from $(F_{p-1})_a^U$ by normalization. It can be checked that G_i for $1 \leq i \leq p$ satisfy the conditions of (2) of the statement of the lemma.

□

Consider a sequence of (nonempty) sets X_i for $1 \leq i \leq p$ such that X_i is either a singleton, or is included in $\bigcup_{j<i} X_j$. We will call such a sequence of sets a normal sequence (of length p). Note that the sequence $X_1,\ldots,X_p$ in Definition A.1 is a normal sequence. Then

A.4 Lemma

Let $X_1,\ldots,X_p$ be a normal sequence of sets with the additional property that each element of $\bigcup_{i=1}^{p} X_i$ appears in at most k elements of the sequence. Then if $p \geq (k+1)^m$, there exists a subset $Y \subseteq \bigcup_{i=1}^{p} X_i$ and an increasing sequence of indices $i(1),i(2),\ldots,i(q)$ such that (1) $q \geq m$,

(2) $i(1) = 1$, (3) $X_{i(j)} \cap Y$ is a singleton for $1 \leq j \leq q$, (4) $X_\ell \cap Y = \phi$ if $\ell \leq i(q)$ and $\ell \neq i(j)$ for $1 \leq j \leq q$, and (5) if $x \in Y$, $j_1 < j_2 < j_3$ and $x \in X_{i(j_1)}$, $x \in X_{i(j_3)}$, then also $x \in X_{i(j_2)}$.

Proof

(this is a direct translation of the proof of Lemma 2 of [Ho68] into our terminology). Let $\bigcup_{i=1}^{p} X_i = \{x_1, x_2, \ldots\}$. Without loss of generality assume that $x_1 \in X_1$. If $m = 1$, set $Y = \{x_1\}$, $i(1) = 1$, and conditions 1-5 are satisfied. For the inductive step two cases are distinguished.

Case I. x_1 occurs in none of the sets X_j, $2 \leq j \leq (k+1)^{m-1} + 1$. Setting $r = (k+1)^{m-1}+1$, the sequence $X_2, \ldots, X_r$ is normal and each element occurs in at most k of the X_j, $2 \leq j \leq r$. If $Z \subseteq \bigcup_{j=2}^{r} X_j$ and the sequence $j(1), \ldots, j(q-1)$ are obtained by the inductive hypothesis, then $\{x_1\} \cup Z = Y$ and $i(1) = 1$, $i(2) = j(1), \ldots, i(q) = j(q-1)$ satisfy conditions 1-5.

Case II. Assume that x_1 occurs in some X_j, $2 \leq j \leq (k+1)^{m-1}+1$ and let h be the smallest such number j. Furthermore, let V be the set of elements different from x_1, and occurring in $X_2, \ldots, X_{h-1}$. Delete the elements of V from $X_1, X_h, \ldots, X_p$, and delete those among $X_1 X_2, \ldots, X_p$ that remain empty. Let the resulting sequence be $Y_1, \ldots, Y_p$. The length of the sequence $(X_1, X_h, X_{h+1}, \ldots, X_p)$ is at least $p-(k+1)^{m-1}+1$.

There are less than $(k+1)^{m-1}$ distinct variables in $X_2, \ldots, X_{h-1}$, each one occurring in at most k-1 of the formulas $X_1, X_h, \ldots, X_p$. Therefore,

$$r \geq p-(k+1)^{m-1}+1-(k-1)(k+1)^{m-1}, \text{i.e.},$$

$$r \geq (k+1)^{m-1}+1$$

The sequence $Y_2,\ldots,Y_r$ is normal and its length is at least $(k+1)^{m-1}$. x_1 occurs in Y_2. Let $Z \subseteq \bigcup_{j=2}^{r} Y_j$ and the sequence $j(1) = 2, j(2),\ldots,$ $(q-1)$ be obtained according to the inductive hypothesis for $Y_2,\ldots,Y_p$. Then Z and $i(1) = 1, i(2) = j(1),\ldots,i(q) = j(q-1)$ (where $q \geq m$), satisfy conditions 1-5. □

Let there be given a (*,*,k)-formula F with the proper nesting sequence $\underline{G} = (G_1,\ldots,G_p)$ such that G_i is a formula over Φ. As has already been remarked above, $X_1,\ldots,X_p$ (see Definition A.1) is a normal sequence of sets.

If $p \geq (k+1)^m$, then by Lemma A.4 there exists a set $Y \subseteq \bigcup_{i=1}^{p} X_i$ and q indices $i(j)$ for $1 \leq j \leq q$ such that conditions 1-5 hold. Note that if $m = k \cdot t$, then $|Y| \geq t$ since no variable appears more than k times in $\underline{G}$ ($\underline{G}$ is proper). In particular, consider only $Z = \{x_1,\ldots,x_t\} \subseteq Y$ where $x_1,\ldots,x_t$ are numbered in the order of their appearance in $\underline{G}$. Note that due to condition 5 of Lemma A.4, if $x, y \in Y$ and y follows x in $\underline{G}$, then x cannot appear again after y in $\underline{G}$. Let G be as defined in Definition A.1. Then we will let the reader convince himself that G_a^Z (hence also F_a^Z) is equivalent to K(Z,G') where K(Z,z) is an e_t-component over Φ^a with input variable z, and G' is a certain formula over Φ^a such that each variable of G' occurs at most k-1 times.

Note that in this case we do not know the size of S(G'). This can be remedied in the following way: There are two cases; either $|S(G')| \geq 1/2 \cdot t$, or not. In the first case perform an a-merger on K with basis S(G'), after which we obtain an SC of an e-component K' of length $\geq 1/2 \cdot t$ and a formula G'' (through the input variable) such that S(G'') equals the set of lateral variables of K'; in the second case perform an a-merger on K(Z,G') with basis Z-S(G') in which case we obtain an e-component K' of length $\geq 1/2 \cdot t$ with a constant input operator.

We summarize the preceding in the following

A.5 Lemma

Let there be given a (*,*,k)-formula F with a proper nesting sequence of length $p \geq 1$ composed of formulas over Φ. Then if $p \geq (k+1)^{2k \cdot t}$, there exists a set $Z \subseteq S(F)$, $|F| \geq t$, and F_a^Z is either equivalent to an SC of an e_t-component K over Φ^a and a formula G over Φ^a such that S(G) is the set of lateral variables of K, and no variable of G occurs more than k-1 times in G; or to an e_t-component K over Φ^a with constant input operator.

Let there be given a PC F of the formulas $F_1,\ldots,F_r$ where $r \leq n_{max}$ such that $|S(F)| = q$ and each variable appears in at least two among $F_1,\ldots,F_r$ (i.e., a situation as described in (1) of the statement of Lemma A.3). We are interested in obtaining a (nonempty) subset $X \subseteq S(F)$ such that when the variables outside of X have been replaced

by the constant a, $|S(norm(F_i)_a^X))|$ for those F_i where not all variables have been replaced with a is equal or larger than a predetermined number t (as large as possible).

We could solve the problem as follows: Each variable of S(F) appears in a certain subset of the formulas $F_1,\ldots,F_r$. The number of such subsets is 2^r (in general, $\leq 2^{n_{max}}$); thus, we are sure to find a subset X with $|X| \geq \frac{q}{n_{max}}$ such that all elements of X appear in the same subset of $F_1,\ldots,F_r^2$.

However, we can improve this number. Let us construct the occurrence table of F. The table consists of rows corresponding to elements of S(F), and of columns corresponding to F_i for $1 \leq i \leq r$. The entry a_{ij} is 1 if x_i occurs in F_j and 0 otherwise. We will try to extract a subset $X \subseteq S(F)$ such that either all variables of F_i are replaced by a, or $S(norm(F_i)_a^X))$ contains $\geq$ elements (t will be determined later).

If all columns in the occurrence table contain $\geq$ t 1's, we are done and X = S(F). Suppose not. Let the column j contain $<$ t 1's. Delete all rows corresponding to the 1's in column j and column j itself. Let the set of variables corresponding to the remaining rows be X_1. Consider the remainder of the occurrence table (i.e., minus the deleted rows and column); and again look for the column with $<$ 1's. If it does not exist, we are done and $X = X_1$. If such a column exists continue. Now two things can happen. Either at some point we end up with a certain subset of columns, all of which contain $\geq$ t 1's, or we end up with two columns that both contain $<$ t 1's. We shall see that by

an appropriate choice of t, the latter case cannot happen. The number of 1's in the whole table $\geq 2q$ (each variable occurs in at least two formulas). The smallest number of 1's remaining after all but two columns have been deleted $\geq 2q-m$ where m is largest possible number of 1's that can be deleted in the course of this procedure. $m = (t-1)\cdot(r+r-1+r-2+\ldots+3) = (t-1)\cdot\frac{(r+3)(r-2)}{2}$ (this corresponds to the case when each deleted row contains only 1's and at each stage t-1 rows are deleted). If, after the table is reduced to two columns, both columns are to contain $\geq t$ 1's (both have to contain the same number of 1's since each variable occurs in at least two formulas), then

$$\frac{2q-m}{2} \geq t$$

or since $r \leq n_{max}$

$$t \leq \frac{4q+c}{c+4} \quad \text{where } c = (n_{max}+3)(n_{max}-2)$$

For large n_{max} this is better than the previous bound. This result can be summarized in

A.6 Lemma

Let there be given a PC F of the formulas $F_1,\ldots,F_r$ over Φ where $r \leq n_{max}$ such that $|S(F)| = q$ and each variable appears in at least two among $F_1,\ldots,F_r$. Then if

$$q \geq \frac{(c+4)t-c}{4} \quad \text{where } t \geq 1$$

we can find a subset $X \subseteq S(F)$ such that F_a^X is equivalent to a PC of the formulas $G_1,\ldots,G_s$ over Φ^a and $S(G_i) \geq t$ for $1 \leq i \leq s \leq r$.

Lemmas A.3, A.5, and A.6 can be combined into

A.7 Lemma

Let F be an (Φ,n,k)-formula. Then for any $t \geq 1$ and $a \in D$ if

$$n \geq \eta_6((k+1)^{2k\cdot t}, \frac{(c+4)\cdot t-c}{4})$$

(see Lemma A.6 for the value of c), there exists a subset $X \subseteq S(F)$ such that either

(1) F_a^X is equivalent to a PC of the formulas $F_1,\ldots,F_r$ over Φ^a where $r \leq n_{max}$, each variable of F_i occurs at most k-1 times in it, and F_i contains at least t variables of X or

(2) F_a^X is equivalent to an SC of an e_t-component K over Φ^a with a formula G over Φ^a (through the input variable) such that S(G) is the set of the lateral variables of K and no variable occurs more than k-1 times in G; or to an e_t-component K over Φ^a with a constant input operator.

A.8 Lemma

Let F be a (Φ,n,k)-formula. Then for any $t \geq 1$ and $a \in D$ if

$$n \geq \eta_7(t,k)$$

then there exists a subset $X \subseteq S(F)$ such that F_a^X is equivalent to an SPCeC over Φ^a G such that (1) G has $\leq n_{max}^k$ components, (2) each component is of length $\geq t$, and (3) the terminal components of G have constant input operators.

Proof

$\eta_7(t,1) = n_{max}^t$. In this case T(F) has at least one branch connected to t+1 variable symbols (k=1 and thus all variable symbols are distinct) at different nodes. This branch can be converted into an e_t-component with constant input operator. The idea is illustrated in Fig. A.1.

$$\eta_7(t,k+1) = \eta_6((k+2)^{2(k+1)\cdot\eta_7(t,k)}, \frac{(c+4)\cdot\eta_7(t,k)-c}{4})$$

We can apply Lemma A.7. The result is either (1) an e-component K of the correct length and constant input operator, (2) an SC of an e-component over Φ^a of the correct length and a formula to which we can apply the inductive hypothesis, and (3) a PC of formulas to which we can apply the inductive hypothesis. In each case we obtain an SPCeC with the desired properties. □

A.9 Lemma

Let F be a (Φ,nk)-formula. Then for any $t \geq 1$ and $a \in D$ if

$$n \geq \eta_8(t,k)$$

there exists a subset $X \subseteq S(F)$ such that F_a^X is equivalent to an SPCeC over Φ^a G such that (1) G has $\leq k$ components, (2) each component has X as the set of its lateral variables, (3) the terminal components have constant input operators and (4) $|X| = t$.

Proof

Set $\eta_8(t,k) = \eta_7(s \cdot t, k)$ where $s = \binom{n_{max}^k}{k} + \binom{n_{max}^k}{k-1} + \ldots + \binom{n_{max}^k}{1}$

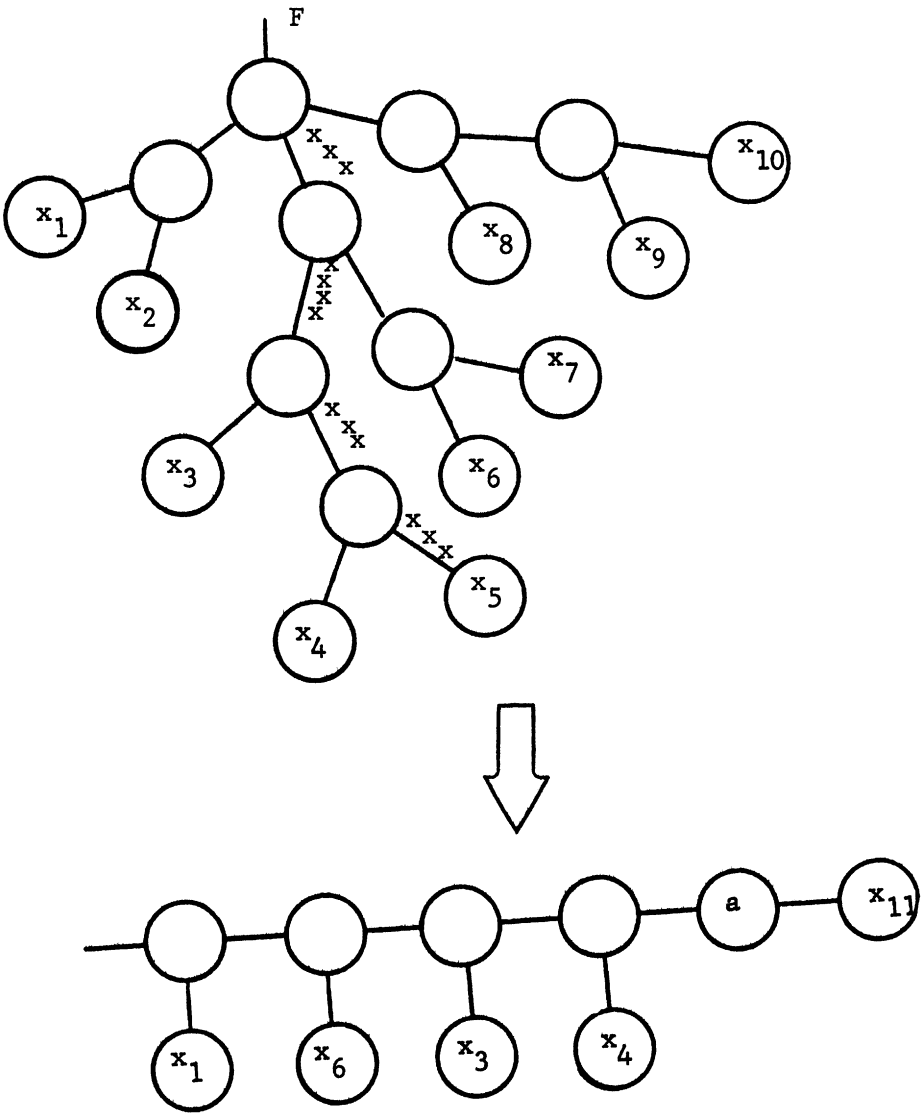
Apply Lemma A.8 to obtain a SPCeC G' with all components having length $\geq s \cdot t$. Since each variable appears at most k times, it can occur in at most k components. s is the number of nonempty subsets of $\leq k$ elements. Thus, if the number of variables is as indicated we are sure to kind in G' a subset of t variables that all occur in the same set of components of G'. After performing an a-merger with this set as basis, we obtain the desired SPCeC G. □

Remarks on the bounds in Lemmas A.3-A.9. If η_6 is approximated by $n_{max}^p \cdot q$, then η_7 is inductively defined as follows:

$$\eta_7(t,1) = n_{max}^t$$

$$\eta_7(t,k) = \gamma \cdot n_{max}^{(k+1)^{2k \cdot \eta_7(t,k-1)}} \cdot \eta_7(t,k-1)$$

for a certain constant γ. Thus we see that $\eta_8(t,k) \geq \text{iexp}(b,2k) = b^{b^{\cdot^{\cdot^{\cdot^{b}}}}}$ (2k times) for $k \geq k(b)$ for any constant b (t has not been included in the estimate because in applications it is constant).



Conversion of a formula F where each variable occurs only once into an equivalent e-component by setting certain variables to a.

Fig. A.1

APPENDIX B

THE LENGTH OF THE MOD 2 SUM OVER Π[†]

There is an isomorphism between the set of formulas over Π and series-parellel contact networks. We assume the reader is familiar with this model as well as with the isomorphism in question. In this case if F is a formula over Π, then L(F) corresponds to the number of contacts in the network corresponding to F.

For convenience, we will derive the result in contact network terminology.

Given a (series-parallel) contact network C, a chain is set of contacts such that when they are all closed, C conducts (we will say "C is 1"); a cut set is a set of contacts such that when they are all open, C does not conduct (we will say "C is 0"). In the obvious way, we define minimal chain, minimal cut set (i.e., when one contact is deleted the corresponding property does not hold).

B.1 Lemma

Given a contact network C and any minimal chain and minimal cut set, their intersection is a singleton.

[†]This result is due to Khrapchenko [Kh71].

Proof

By induction on the number m of contacts in C. For m = 1 the assertion is obviously true. If $m > 1$, C must be either a series combination of smaller networks C_1 and C_2, or a parallel combination of smaller networks C_1 and C_2. In each case it is simple to establish the lemma. □

Suppose now we have a contact network S that represents $\bigoplus_{i=1}^{n} x_i$. Let m_j denote the number of contacts labeled with x_j or $\bar{x}_j$. Then we are interested in $\sum_{j=1}^{n} m_j$.

Consider n-tuples (a_i) for $1 \leq i \leq n$ and $a_i \in \{0,1\}$. An n-tuple of this kind will be called even if it has an even number of 1's, otherwise it is odd. Obviously S must be 1 on odd n-tuples and 0 on even ones.

Consider an arbitrary odd n-tuple $\underline{a} = (a_1,\ldots,a_i,\ldots,a_n)$ and an even n-tuple $\underline{b} = (b_1,\ldots,b_i,\ldots,b_n)$ at Hamming distance 1 from $\underline{a}$. If $b_i = \bar{a}_i$, then all other components of $\underline{a}$ and $\underline{b}$ are equal. $\underline{e}_i$ will denote the n-tuple with a single 1 in the ith place. Then we will write $\underline{b} = \underline{a} \oplus \underline{e}_i$.

To each odd n-tuple $\underline{a}$ we can assign a minimal chain $c(\underline{a})$ (consisting of a subset of contacts of S that are closed at $\underline{a}$ and that do form a minimal chain); similarly, to each even n-tuple $\underline{b}$ we can assign a minimal cut set $s(\underline{b})$ (consisting of a set of contacts of S that are open at $\underline{b}$ and that do form a minimal cut set).

Let $\underline{a}$ be odd, $\underline{b} = \underline{a} \oplus \underline{e}_i$ even. Then by Lemma B.1, $c(\underline{a}) \cap s(\underline{b})$ is a singleton; in fact, it is easy to verify that it must be a contact labeled either with x_i or $\bar{x}_i$.

We build now Tables I and II. The rows of Table I correspond to odd n-tuples while those of Table II correspond to even n-tuples. Thus both have 2^{n-1} rows. The columns of both tables correspond to the variable x_i for $1 \leq i \leq n$. The entry $\alpha(\underline{a},j)$ in Table I is $c(\underline{a}) \cap s(\underline{a} \oplus \underline{e}_j)$. This entry will be represented by a number between 1 and m_j.

Let t_{ij} denote the number of times contact number i (among those labeled with x_j or $\bar{x}_j$) appears in column j of Table I. Then

$$\sum_{i=1}^{m_j} t_{ij} = 2^{n-1} \tag{B.1}$$

The entry $\beta(\underline{b},j)$ of Table II is $s(\underline{b}) \cap c(\underline{b} \oplus \underline{e}_j)$. (B.1) again holds.

Construct now Table III. The rows of Table III correspond to all possible pairs $(\underline{a},\underline{b})$ where $\underline{a}$ and $\underline{b}$ are odd and even n-tuples respectively. The columns of Table III again correspond to variables. An entry of Table III is $\gamma(\underline{a},\underline{b},j) = (\alpha(\underline{a},j), \beta(\underline{b},j))$.

Consider now the diagonal entries of Table III (i.e., (α,β) such that $\alpha = \beta$). Let $(\alpha(\underline{a},j), \beta(\underline{b},j))$ be such an entry. Then $\alpha(\underline{a},j) = \beta(\underline{b},j) = c(\underline{a}) \cap s(\underline{b})$. Thus, by Lemma B.1, there can be only one such entry in a row.

The number of diagonal entries is $\sum_{j=1}^{n} \sum_{i=1}^{m_j} t_{ij}^2$. Thus,

$$\sum_{j=1}^{n} \sum_{i=1}^{m_j} t_{ij}^2 \leq 2^{2(n-1)} \tag{B.2}$$

Combining a version of Cauchy's inequality

$$\frac{1}{m_j} \left(\sum_{i=1}^{m_j} t_{ij} \right)^2 \leq \sum_{i=1}^{m_j} t_{ij}^2$$

with (B.1) and (B.2), we obtain

$$\sum_{j=1}^{n} \frac{1}{m_j} \leq 1$$

This time we apply the inequality

$$\left(\sum_{j=1}^{n} m_j \right) \cdot \left(\sum_{j=1}^{n} \frac{1}{m_j} \right) \geq n^2$$

(for both inequalities see, e.g., [Mt64]p. 9), and obtain the desired result

$$\sum_{j=1}^{n} m_j \geq n^2$$

LITERATURE

Ar69 M. A. Arbib, Theories of Abstract Automata, Prentice-Hall, 1969.

Av69 A. Avizienis, On the Problem of Computational Time and Complexity of Arithmetic Functions, Proc. ACM Symposium Theory of Computing, May 5-7, 1969, Marina del Rey, California, pp. 255-258.

Be71 C. Berge, Principles of Combinatorics, Academic Press, 1971.

Co71 S. A. Cook, The Complexity of Theorem Proving Procedures, Proc. 3rd Ann. Symposium Theory Computing, Shaker Heights, Ohio, May 3-5, 1971 pp. 151-158.

Gr59 Editors, Eugene M. Grabbe et al., Handbook of Automation Computation and Control, Vol. 2, John Wiley, 1959.

Ha71 L. H. Harper and J. E. Savage, On the Complexity of the Marriage Problem (unpublished) .

Ho68 L. Hodes and E. Specker, Lengths of Formulas and Elimination of Quantifiers I, Contributions to Mathematical Logic, K. Schutte, editor, North Holland Publ. Co., 1968, pp. 175-188.

Ho70 L. Hodes, The Logical Complexity of Geometric Properties in the Plane, Journal ACM, 17, No. 2, pp. 339-347.

Kh71 V. M. Khrapchenko, On the Complexity of the Realization of the Linear Function in the Class of π-Circuits, Mat. Zametki, 9, No. 1, 1971, pp. 35-40 (Russian).

Kr59 R. E. Krichevskii, Realization of Functions by Superpositions, Prob. Cypernetics II, 1961, pp. 458-477, Pergamon Press (translated from the Russian).

La67 S. Lang, Algebra, Addison-Wesley, 1967.

Lu59 O. B. Lupanov, Complexity of Formula Realizations of Functions of Logical Algebra, Prob. Cybernetics III, A. A. Lyapunov, editor, Pergamon Press, 1962, pp. 782-811 (translated from the Russian).

Lu70 O. B. Lupanov, On Some Results in the Mathematical Theory of Synthesis of Control Systems, Information Materials 5(42), Ac. Sci. USSR, Moscow 1970, pp. 16-22 (Russian).

Mi69 M. Minsky and S. Papert, Perceptrons, MIT Press, 1969.

Mk71 R. McKenzie, et al. On Boolean Functions and Connected Sets, Math. Systems Theory, 5, No. 3, pp. 259-270.

Mt64 D. S. Mitrinovic, Elementary Inequalities, P. Noordhoff Ltd. Groningen, 1964.

My71 J. P. Mylopoulos and T. Pavlidis, On the Topological Properties of Quantized Spaces I, II, Journal ACM, 18, No. 2, pp. 239-254.

Ne 66 E. I. Neciporuk, A Boolean Function, Soviet Math. Dokl., 2, No. 4, 1966, pp. 999-1000.

Ri42 J. Riordan, C. E. Shannon, The Number of Two Terminal Series-Parallel Networks, J. Math. and Phys. 21, 1942, pp. 83-93.

Ry63 H. J. Ryser, Combinatorial Mathematics, MAA Math. Monographs, No. 14, John Wiley, 1963.

Sh49 C. E. Shannon, The Synthesis of Two-Terminal Switching Circuits, Bell System Tech. J., 28, No. 1, 1949, pp. 59-98.

Su61 B. A. Subbotovskaya, Realizations of Linear Functions by Formulas Using $\vee$, &, -, Soviet Math. Dokl., 2, No. 2, 1961, pp. 110-112.

Vi70 B. Vilfan, Cyclic Perceptrons and Pattern Counting Machines, Proc. 4th Ann. Princeton Conf. Info. Sci. and Syst., Princeton U., March 1970

Ya54 S. V. Yablonskii, The Realization of the Linear Function in the Class of π-Circuits, Dokl, Ac. Sci. USSR, 94, No. 5, pp. 805-806 (Russian).

Ya59 S. V. Yablonskii, On the Impossibility of Eliminating Exhaustive Search of Boolean Functions in the Solution of Some Problems in the Theory of Circuits, Dokl. Ac. Sci. USSR, 124, No. 1, pp. 44-47, (Russian).

This empty page was substituted for a blank page in the original document.

CS-TR Scanning Project
Document Control Form

Date : 1/23/96

Report # LCS-TR-97

Each of the following should be identified by a checkmark:
Originating Department:

☐ Artificial Intellegence Laboratory (AI)
☒ Laboratory for Computer Science (LCS)

Document Type:

☒ Technical Report (TR) ☐ Technical Memo (TM)
☐ Other:_____

Document Information

Number of pages: 118 (124 - IMAGES)
Not to include DOD forms, printer intstructions, etc... original pages only.

Originals are:
☐ Single-sided or
☒ Double-sided

Intended to be printed as :
☐ Single-sided or
☒ Double-sided

Print type:
☐ Typewriter ☐ Offset Press ☐ Laser Print
☐ InkJet Printer ☒ Unknown ☐ Other:_____

Check each if included with document:
☒ DOD Form ☐ Funding Agent Form ☒ Cover Page
☐ Spine ☐ Printers Notes ☐ Photo negatives
☐ Other: _____

Page Data:

Blank Pages(by page number): FOLLOWS LAST PAGE

Photographs/Tonal Material (by page number):_____

Other (note description/page number):

Description :	Page Number:
IMAGE MAP:	(1 - 118) UNNUMBERED TITLE PAGE, 2-117, UNNUMBERED BLANK
	(119-124) SCANCONTROL, COVER, DOD, TARGETS (3)

Scanning Agent Signoff:
Date Received: 1/23/96 Date Scanned: 1/26/96 Date Returned: 2/1/96

Scanning Agent Signature: Michael W. Cook

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* Massachusetts Institute of Technology Project MAC	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP NONE

3. REPORT TITLE
The Complexity of Finite Functions

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*
Ph.D., Department of Electrical Engineering, February 1972

5. AUTHOR(S) *(Last name, first name, initial)*
Vilfan, Bostjan

6. REPORT DATE March 1972	7a. TOTAL NO. OF PAGES 118	7b. NO. OF REFS 25
8a. CONTRACT OR GRANT NO. N00014-70-A-0362-0001 b. PROJECT NO. c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) MAC TR-97 (Thesis) 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)* NONE	

10. AVAILABILITY/LIMITATION NOTICES
Distribution of this document is unlimited

11. SUPPLEMENTARY NOTES None	12. SPONSORING MILITARY ACTIVITY Advanced Research Projects Agency 3D-200 Pentagon Washington, D. C. 20301

13. ABSTRACT The topics covered are the length of formulas for finite functions, the order of cyclic perceptrons, and pattern counting machines. Using a generalization of a theorem of Specker, it is shown that the Boolean function is 1 if the number mod p of arguments equal to 1 is 0 cannot be represented by a formula of length proportional to the number of arguments if k-ary logic is used with p>k. The same thing can be shown for arbitrary k if the only binary operators used are max(x,y) and min(x,y). It is also shown that the connectivity predicate cannot be represented by a formula of this kind, regardless of k and of the operators used. Next shown is that the connectivity predicate and the Euler number predicate cannot be represented by finite order cyclic perceptrons. Finally, it is shown that the only topological predicates that can be reconstructed from the k-subpattern spectrum of a given square pattern of 0's and 1's are functions of the Euler number. The k-subpattern spectrum of a pattern is a tuple given the number of occurrences of any kxk square subpattern in the original pattern.

14. KEY WORDS
computational complexity
combinatorics
finite functions

DD FORM 1 NOV 68 1473 (M.I.T.)

UNCLASSIFIED
Security Classification

Scanning Agent Identification Target

Scanning of this document was supported in part by the **Corporation for National Research Initiatives,** using funds from the **Advanced Research Projects Agency** of the **United states Government** under Grant: **MDA972-92-J1029.**

The scanning agent for this project was the **Document Services** department of the **M.I.T Libraries.** Technical support for this project was also provided by the **M.I.T. Laboratory for Computer Sciences.**

Scanned

Date: 1/26/1996

M.I.T. Libraries
Document Services