MIT/LCS/TM-55

A CLASS OF BOOLEAN FUNCTIONS WITH LINEAR COMBINATIONAL COMPEXITY

W. N. Hsieh
L.H. Harper
J.E. Savage

October 1974

TM-55

A Class of Boolean Functions with Linear Combinational Complexity

By

W.N. Hsieh,* L.H. Harper† and J.E. Savage††

September 1974

* Department of Mathematics, M.I.T., Cambridge, Massachusetts 02139
Network Analysis, Beechwood, Old Tappan Road, Glen Cove, New York 11542

† Department of Mathematics, University of California, Riverside, California 92501

†† Division of Engineering, Brown University, Providence, Rhode Island 02912
(During the preparation of this paper this author was visiting the Department of Mathematics, Technological University Eindhoven with the support of a Guggenheim Fellowship and a Fulbright-Hays Award.)

Research reported here was supported by the National Science Foundation under research grant GJ-34671 at M.I.T., Project MAC, and under grant GJ-32270 at Brown University.

Abstract

In this paper we investigate the combinational complexity of Boolean functions satisfying a certain property, $\wp^n_{k,m}$. A function of n variables has the $\wp^n_{k,m}$ property if there are at least m functions obtainable from each way of restricting it to a subset of n-k variables. We show that the complexity of a $\wp^n_{3,5}$ function is no less than $\frac{7n-4}{6}$, and this bound cannot be much improved. Further, we find that for each k, there are $\wp^n_{k,2^k}$ functions with complexity linear in n.

I. Introduction

The size of combinational networks, or equivalently the length of straight-line programs, provides a measure of complexity for Boolean functions which reflects the difficulty of computing them (cf. [Sa72]). A well-known result due to Shannon [Sh49] and Lupanov [Lu58] establishes that almost all (in a precise sense which we shall not describe) Boolean functions of n variables have combinational complexity assympototic to $2^n/n$. Ehrenfeucht [Eh72], Meyer [Me74a] and Stockmeyer [St74] have shown recently that particular Boolean functions which encode finite portions of a variety of decision problems from mathematical logic and automata theory have exponential combinational complexity. However, the proof technique used there does not appear likely to yield lower bounds on the combinational complexity of functions whose complexity is bounded by polynomials in the number of variables.

In this paper we investigate a property which reflects a way in which a function depends on subsets of its variables, and we obtain small but non-trivial linear lower bounds on the combinational complexity of functions with the property. Similar properties have recently been investigated by Schnorr [Sc74], who also obtains small linear lower bounds on combinational complexity, and by Neciporuk [Ne66], who obtains roughly quadratic lower bounds on the size of Boolean formulas.

A function of n variables has the $\wp^n_{k,m}$ property if there are at least m different functions obtainable from each way of restricting it to a subset of n-k variables. (Precise definitions appear in Section II, below.) Let $c(\wp^n_{k,m})$ be the least number of two input gates sufficient to construct combinational networks for one of the functions in $\wp^n_{k,m}$.

In Section III we prove $c(\wp^n_{1,2}) = n-1$ and $c(\wp^n_{2,3}) = n$. In Section IV we show $c(\wp^n_{3,5}) \geq \frac{7n-4}{6}$ by deriving simple structural constraints on networks computing $\wp^n_{3,5}$ functions and then translating these constraints into a linear programming problem. In Section V we show that $c(\wp^n_{3,5}) \leq (20n-1)/17$ by exhibiting networks for $\wp_{3,5}$ functions. Since $\frac{7n-4}{6} \approx 1.167n$ and $\frac{20n-1}{17} \approx 1.176n$, the bounds are fairly close. In Section VI we consider the $\wp_{k,2^k}$ functions. We present a simple $\wp_{k,2^k}$ function due to M. Rabin. Using this function as the basis, we then show constructively that there are infinitely many $\wp_{k,2^k}$ functions with linear complexity.

II. Definitions and Preliminaries

We review the formal definitions of combinational networks and the functions they compute.

A binary combinational network, or simply a network, is a directed node-labelled acyclic graph $\mathcal{N}$ such that

(i) each node of $\mathcal{N}$ has in-degree either 0 or 2, and arcs entering a node of in-degree two are ordered so we may speak of the first and second input arcs of a node,

(ii) each node of $\mathcal{N}$ with in-degree two is labelled with a Boolean function of two arguments, and each node with in-degree zero is labelled with a distinct variable, and

(iii) there is a unique node with out-degree 0 which is called the output node of $\mathcal{N}$.

A node φ with in-degree two is called a gate. The node in $\mathcal{N}$ that connects to φ through the first (resp. second) input arc to φ is called the first (resp. second) input node to φ. Similarly, the out-going arcs from a node φ

are called the output arcs from φ, and the nodes that have φ as an input node are called the output nodes of φ. Suppose φ and ψ are nodes in $\mathcal{N}$ such that there is a directed path in $\mathcal{N}$ from φ to ψ, then φ is called a source node to ψ, and ψ is called a successor node to φ. We consider φ to be a source node and a successor node to itself. A gate with out-degree k is represented schematically in Figure 1 with the associated Boolean function written inside the half-disc.

Figure 1. A gate with fan-out k

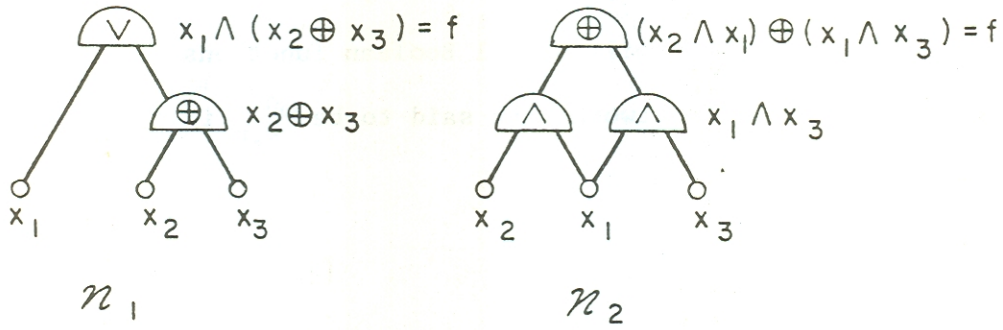
A node with in-degree 0 is called a variable node. A variable node with out-degree k is represented schematically in Figure 2.

Figure 2. A variable node with fan-out k

In a network with variable nodes labelled $x_1,\ldots,x_n$, the variable node labelled with variable x_i is said to compute the projection function $U_i^n(x_1,\ldots,x_n) = x_i$. Proceeding inductively, a gate φ labelled with a

Boolean function h of two arguments is said to compute the function $f_\varphi(x_1,\ldots,x_n) \equiv h(f_{\varphi_1}(x_1,\ldots,x_n), f_{\varphi_2}(x_1,\ldots,x_n))$ where φ_1, φ_2 and f_{φ_1}, f_{φ_2} are the first and second input nodes to φ and the functions they compute. The network as a whole will be said to compute the function associated in this way with the output node.

For example, the network $\mathcal{N}_1$ in Figure 3 computes the function $f(x_1,x_2,x_3) = x_1 \wedge (x_2 \oplus x_3)$. (We use $\oplus$ to denote sum modulo 2.) Note that the network $\mathcal{N}_2$ also computes f.



$\mathcal{N}_1$ $\mathcal{N}_2$

Figure 3. Two networks for $x_1 \wedge (x_2 \oplus x_3)$

The combinational complexity, $c(\mathcal{N})$, of a network $\mathcal{N}$, is the total number of gates in $\mathcal{N}$. The combinational complexity, c(f), of a finite Boolean function f, is the minimum of $c(\mathcal{N})$, where $\mathcal{N}$ ranges over all networks computing f. Also, the combinational complexity, $c(\mathcal{F})$, of a family $\mathcal{F}$ of Boolean functions is the minimum of c(f), where f ranges over all functions in $\mathcal{F}$.

Let $X = (x_1,\ldots,x_k)$ be a k-tuple of variables. $A = (a_1,\ldots,a_k)$ with $a_i \in \{0,1\}$ is called an assignment of X. We also use $A(x_i)$ to denote a_i, the value assigned to x_i by A. For a Boolean function f, f_A^X is the function of the variables not in X obtained from f by setting each $x_i \in X$ to a_i. We use $\{f^X\}$ to denote $\{f_A^X : A \text{ is an assignment of } X\}$, and $\{f_B^{Y,X}\}$ to denote the

family $\{f^{Y,X}_{B,A}$: A is an assignment of X$\}$. (This latter notation is used only when X and Y are disjoint.) We say f depends on x if $f^x_0 \neq f^x_1$. V(f) is the set of variables upon which f depends.

Definition: A Boolean function f is said to have the $\wp^n_{k,m}$ property if $n \geq k$, $|V(f)| = n$, and for every set X of k variables in V(f), $|\{f^X\}| \geq m$.†

Note that f has the $\wp^n_{1,2}$ property if and only if $|V(f)| = n$.

We shall use '$f \in \wp^n_{k,m}$' or 'f is $\wp^n_{k,m}$' to mean f has the $\wp^n_{k,m}$ property. $\wp^n_{k,m}$ will denote the family of all Boolean functions with the $\wp^n_{k,m}$ property and $\wp_{k,m} = \bigcup_{n \geq k} \wp^n_{k,m}$. Networks are said to be $\wp^n_{k,m}$ if they compute functions with this property.

Lemma 2.1.†† $f \in \wp^n_{k,m} \Rightarrow f \in \wp^n_{k-1,\lceil m/2 \rceil}$.

Proof: Let f be a Boolean function with $|V(f)| = n$. If $f \notin \wp^n_{k-1,\lceil m/2 \rceil}$, then there is a set Y of k-1 variables in V(f) such that $|\{f^Y\}| \leq \lceil m/2 \rceil - 1$. Hence, $|\{f^{X,Y}_A\}| \leq \lceil m/2 \rceil - 1$ for any set X of variables and assignment A of X. But then for any x in V(f) - Y, $\{f^{x,Y}\} = \{f^{x,Y}_0\} \cup \{f^{x,Y}_1\}$ has at most $2(\lceil m/2 \rceil - 1) \leq m-1$ members, i.e. $f \notin \wp^n_{k,m}$. Q.E.D.

---

† $|S|$ denotes the cardinality of a set S.

†† $\lceil z \rceil$ denotes the least integer greater than or equal to a number z.

The Weak Duality Theorem in linear programming will be used in Section IV to obtain a complexity lower bound for $\mathcal{P}^n_{3,5}$ functions. For the reader's convenience we state the Theorem below.

Suppose the primal problem ($\mathcal{P}$) is to find real values for $x_1,\ldots,x_n$ which

$$\text{minimize} \quad z = \sum_{j=1}^{n} c_j x_j ,$$

$$\text{subject to} \quad \sum_{j=1}^{n} a_{ij} x_j = b_i \quad \text{for } i \in E,$$

$$\sum_{j=1}^{n} a_{ij} x_j \geq b_i \quad \text{for } i \in \bar{E} ,$$

$$E \cup \bar{E} = \{1,\ldots,m\} ,$$

$$\text{and} \quad x_i \geq 0 \quad \text{for } i \in P,$$

$$x_i \text{ unconstrained in sign for } i \in \bar{P},$$

$$P \cup \bar{P} = \{1,\ldots,n\} .$$

Then the dual ($\mathcal{D}$) to ($\mathcal{P}$) is to find real values for $y_1,\ldots,y_m$ which

$$\text{maximize} \quad v = \sum_{i=1}^{m} y_i b_i$$

$$\text{subject to} \quad \sum_{i=1}^{m} y_i a_{ij} = c_j , \quad j \in \bar{P},$$

$$\sum_{i=1}^{m} y_i a_{ij} \leq c_j , \quad j \in P,$$

and $y_i \geq 0$ for $i \in \bar{E}$, y_i unconstrained in sign for $i \in E$.

Theorem (Weak Duality) If $\bar{x} = (\bar{x}_1, \ldots, \bar{x}_n)$ and $\bar{y} = (\bar{y}_1, \ldots, \bar{y}_m)$ are feasible solutions to ($\mathcal{P}$) and ($\mathcal{D}$), then

$$\bar{z} = \sum_{j=1}^{n} c_j \bar{x}_j \geq \sum_{i=1}^{m} \bar{y}_i b_i = \bar{v} .$$

For a reference on duality in linear programming, see any standard text on linear programming, for example, Lasdon [La70].

III. Values for $c(\mathcal{P}^n_{1,2})$ and $c(\mathcal{P}^n_{2,3})$

In this section we investigate $c(\mathcal{P}^n_{1,2})$ and $c(\mathcal{P}^n_{2,3})$. By simple combinatorial arguments, we show that $c(\mathcal{P}^n_{1,2}) \geq n-1$ and $c(\mathcal{P}^n_{2,3}) \geq n$. We then demonstrate by construction that these bounds are indeed realizable.

Lemma 3.1. (i) $f \in \mathcal{P}^n_{1,2} \Rightarrow c(f) \geq n-1$.

(ii) $f \in \mathcal{P}^n_{2,3} \Rightarrow c(f) \geq n$.

Proof: Let f be a Boolean function with $|V(f)| = n$. Let $\mathcal{N}$ be a network computing f, $|G|$ the number of gates in $\mathcal{N}$ and $|V|$ the number of variable nodes in $\mathcal{N}$. Obviously $|V| \geq n$.

Note that the total number of input arcs in $\mathcal{N}$ = the total number of output arcs in $\mathcal{N}$. Also, every gate in $\mathcal{N}$ has two input arcs, so the total number of input arcs in $\mathcal{N} = 2|G|$.

(i) Suppose $f \in \mathcal{P}^n_{1,2}$. By definition every node except the output node of $\mathcal{N}$ has out-degree ≥ 1. Thus the total number of output arcs in $\mathcal{N} \geq$ (the total number of nodes in $\mathcal{N}$) $- 1 = |G| + |V| - 1 \geq |G| + n - 1$. Thus $2|G| \geq |G| + n - 1$, and hence $c(f) \geq |G| \geq n - 1$.

(ii) Suppose $f \in \mathcal{P}^n_{2,3}$. Then $\mathcal{N}$ cannot have a "subnetwork" of the form in Figure 4.

Figure 4. A Forbidden subnetwork for $\mathcal{P}_{2,3}$.

That is, if two variable nodes are inputs to the same gate, then at least one of them is also an input to some other gate. For if x_i, x_j are input nodes to the gate φ only, then we can obviously express $f(x_1,\ldots,x_n)$ as $h(f_\varphi(x_i,x_j), y_1,\ldots,y_{n-2})$ where $h(z,y_1,\ldots,y_{n-2})$ is some Boolean function of n-1 variables, f_φ is the function labelling gate φ, and $y_1,\ldots,y_{n-2}$ are in $\{x_1,\ldots,x_n\} - \{x_i,x_j\}$. Hence $\{f^{x_i,x_j}\} \subset \{h^z\} = \{h^z_0, h^z_1\}$ and has cardinality at most two, contradicting the assumption that $f \in \mathcal{P}_{2,3}$.

Now since networks are acyclic, there is in any network one gate both of whose input nodes are variables. Thus since $\mathcal{N}$ cannot have a subnetwork of

the form in Figure 4, there is at least one variable node in $\mathcal{N}$ with out-degree two. Hence the total number of output arcs in $\mathcal{N} \geq$ (total number of gates) + (total number of variable nodes + 1) - 1 = $|G| + |V| \geq |G| + n$.

Therefore $2|G| \geq |G| + n$, and so $c(f) \geq |G| \geq n$. Q.E.D.

Theorem 3.1. $c(\mathcal{P}^n_{1,2}) = n - 1$.

Proof: $\bigoplus_{i=1}^{n} x_i$ is obviously $\mathcal{P}^n_{1,2}$ and realizable with n-1 gates labelled with the function $\oplus$. Q.E.D.

We now proceed to develop the upper bounds on $c(\mathcal{P}_{2,3})$.

Lemma 3.2. Let h be a Boolean function, let g(x,y) be a Boolean function which depends on the variable y, and suppose f = g(x,h) with $x \notin V(h)$. Further, suppose X is a set of variables in V(h) and A_1, A_2 are assignments of X such that $h^X_{A_1} \neq h^X_{A_2}$. Then $f^X_{A_1} \neq f^X_{A_2}$.

Proof: Since $h^X_{A_1} \neq h^X_{A_2}$, we can by symmetry suppose that there is an assignment B of V(h) - X such that $h^{X,V(h)-X}_{A_1,\ B} = 0$ and $h^{X,V(h)-X}_{A_2,\ B} = 1$. But then $f^{X,V(h)-X}_{A_1,\ B} = g(x,0) \neq g(x,1) = f^{X,V(h)-X}_{A_2,\ B}$ since g(x,y) depends on y.

Hence $f^X_{A_1} \neq f^X_{A_2}$. Q.E.D.

Lemma 3.3. Suppose f is $\mathcal{P}_{2,3}$ such that $f^y_a \neq 0$ for any variable y and constant a. Let x be a variable not in V(f). Then $x \wedge f$ is also $\mathcal{P}_{2,3}$ with the property that $(x \wedge f)^y_0 \neq \overline{(x \wedge f)}^{\,y}_1$ for any variable y.

Proof: First we show that $(x \wedge f)^y_0 \neq \overline{(x \wedge f)}^y_1$ for any variable y. If y = x, then $\overline{(x \wedge f)}^x_1 = \bar{f}$, while $(x \wedge f)^x_0 = 0$. But $\bar{f} \neq 0$ because f is $\mathcal{P}_{2,3}$. If $y \neq x$, then $(x \wedge f)^{y,x}_{0,0} = 0$ while $\overline{(x \wedge f)}^{y,x}_{1,0} = 1$. Hence again $(x \wedge f)^y_0 \neq \overline{(x \wedge f)}^y_0$.

Now we show that $x \wedge f$ is $\mathcal{P}_{2,3}$. Let X be a set of two variables in $V(x \wedge f)$; we show that $|\{(x \wedge f)^X\}| \geq 3$. If $X \subset V(f)$, then $|\{f^X\}| \geq 3$ because f is $\mathcal{P}_{2,3}$, and hence by Lemma 3.2, $|\{(x \wedge f)^X\}| \geq 3$.

If $X = \{x,y\}$ for some $y \in V(f)$, then $\{(x \wedge f)^X\} = \{f^y_0, f^y_1, 0\}$. Now $y \in V(f)$, so $f^y_0 \neq f^y_1$. Also, by assumption, f^y_0 and f^y_1 are non-zero. Hence there are three distinct members of $\{(x \wedge f)^X\}$. Q.E.D.

Lemma 3.4. Suppose f is $\mathcal{P}_{2,3}$ with the additional property that $f^y_0 \neq \bar{f}^y_1$ for any variable y. Let x be a variable not in V(f). Then $x \oplus f$ is also $\mathcal{P}_{2,3}$ such that $(x \oplus f)^y_a \neq 0$ for any variable y and constant a.

Proof: First note that $(x \oplus f)^y_a \neq 0$ for any variable y and constant a because f is nonconstant and so in order to set $x \oplus f$ to a constant we have to set x and at least one variable in f to constants.

Now we show that $x \oplus f$ is $\mathcal{P}_{2,3}$. Let X be a set of two variables in $V(x \oplus f)$; we show $|\{(x \oplus f)^X\}| \geq 3$.

If $X \subset V(f)$, then $|\{f^X\}| \geq 3$ by assumption. Thus by Lemma 3.2 $|\{(x \oplus f)^X\}| \geq 3$.

If $X = \{x,y\}$ for some $y \in V(f)$, then $\{(x \oplus f)^X\} = \{f^y\} \cup \{\bar{f}^y\}$. But $|\{f^y\}| = |\{\bar{f}^y\}| = 2$ since $y \in V(f)$, and $\{f^y\} \cap \{\bar{f}^y\} = \emptyset$ by assumption. Thus $|\{(x \oplus f)^X\}| = 4$ in this case. Q.E.D.

Theorem 3.2. $c(\mathcal{P}^n_{2,3}) = n$ for $n \geq 3$.

Proof: By Lemma 3.1, it suffices to exhibit a $\mathcal{P}^n_{2,3}$ function realizable in n gates for every $n \geq 3$.

Let f_3 be the function computed by the network in Figure 5.

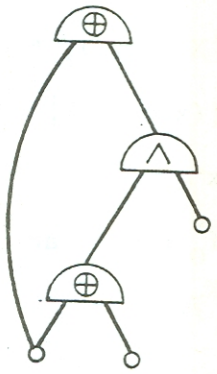


Figure 5. A $\mathcal{P}^3_{2,3}$ network

It is easy to verify that f_3 is $\mathcal{P}^3_{2,3}$ and $f^y_a \neq 0$ for any variable y and constant a.

For $n \geq 2$, define $f_{2n} = f_{2n-1} \wedge x$ where x is a new variable not in $V(f_{2n-1})$, and $f_{2n+1} = f_{2n} \oplus y$ where y is a new variable not in $V(f_{2n})$. Then by induction using Lemmas 3.3 and 3.4, it is obvious that f_n is $\mathcal{P}^n_{2,3}$ and $c(f_n) \leq n$ for all $n \geq 3$.

Q.E.D.

IV. A Lower Bound for $c(\mathcal{P}^n_{3,5})$.

Let f be a $\mathcal{P}^n_{3,5}$ function, $\mathcal{N}$ a minimum gate network computing f, V the set of variable nodes in $\mathcal{N}$ and G the set of gates in $\mathcal{N}$. Assume f depends on all its arguments so $V = V(f)$ and $|V| = n$. Note that for any gate φ, the two input arcs to φ are from different nodes, for otherwise with appropriate

modification to the Boolean functions associated with the other gates, we can eliminate φ, and $\mathcal{N}$ would not have had a minimum number of gates.

We classify the gates in $\mathcal{N}$ into three types:

(1) φ is of $A_k^{p,q}$-type, or $\varphi \in A_k^{p,q}$, where $p \geq 1$, $q \geq 1$, $k \geq 0$, if φ has out-degree k, and the first and second input nodes to φ are variable nodes with out-degree p and q, respectively. With appropriate modification to φ if necessary, we can suppose that $p \geq q$ (if $p < q$, we can use φ' to replace φ, where $\varphi'(a,b) = \varphi(b,a)$; the resulting network still computes f).

(2) φ is of B_k^p-type, or $\varphi \in B_k^p$, where $p \geq 1$ and $k \geq 0$, if φ has out-degree k, one of the input nodes to φ is a variable node with out-degree p and the other input node is a gate. By the same reasoning as in (1), we can suppose that the variable input node is the first input node.

(3) φ is of C_k-type, or $\varphi \in C_k$, where $k \geq 0$, if φ has out-degree k and both input nodes to φ are gates.

The $A_k^{p,q}$, B_k^p and C_k gates are illustrated schematically in Figure 6.

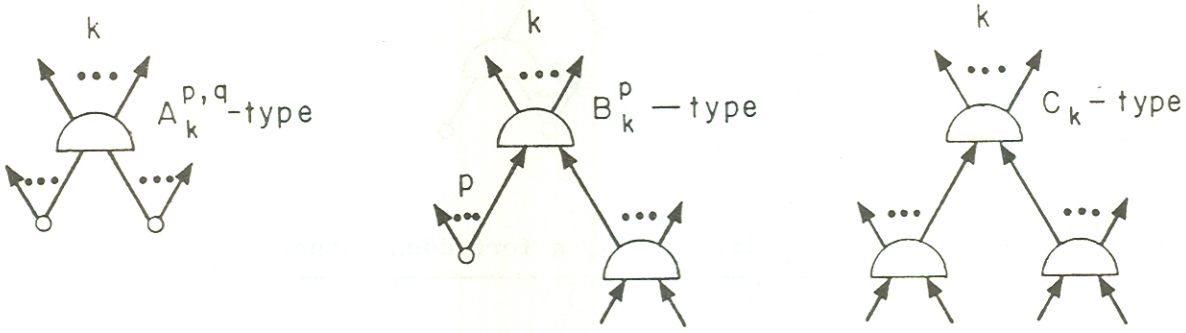


Figure 6. Three gate types

Lemma 4.1. The following restrictions on the local structure of $\mathcal{N}$ must hold:

(a) No $A_k^{1,1}$-type gate occurs in $\mathcal{N}$, i.e. no sub-network of the form in Figure 4.

(b) If a B_1^1-type gate φ is an input node to another B_1^1-type gate ψ, then ψ cannot be an input node to B_k^1-type gates, for $k \geq 0$, i.e. no subnetwork of the form in Figure 7.

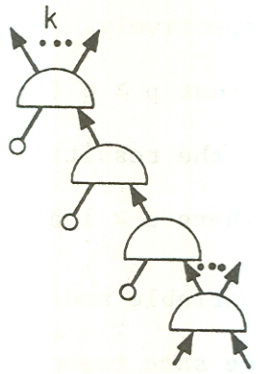


Figure 7. Condition (b), a forbidden subnetwork for $\mathcal{P}_{3,5}$

(c) An $A_1^{p,1}$-type gate cannot be the input node to a B_k^1-type gate, i.e., no subnetwork of the form in Figure 8.

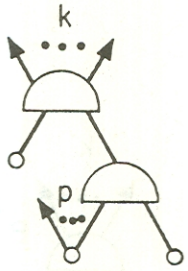


Figure 8. Condition (c), a forbidden subnetwork for $\mathcal{P}_{3,5}$

(d) A variable node of out-degree 2 cannot be an input node of both an $A_k^{2,1}$-type gate and an $A_j^{2,1}$-type gate for $j \geq 0$, $k \geq 0$, i.e. no subnetwork

of the form in Figure 9.

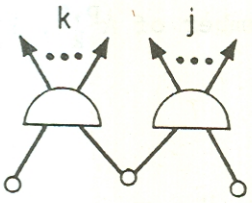


Figure 9. Condition (d), a forbidden subnetwork for $\mathcal{P}_{3,5}$

Proof:

In Lemma 3.1 (ii) we proved that condition (a) was satisfied by $\mathcal{P}_{2,3}$ networks. But $\mathcal{P}_{3,5}$ implies $\mathcal{P}_{2,3}$ by Lemma 2.1, so (a) is proved. We now prove (c), leaving the similar proofs of (b) and (d) to the reader.

Suppose φ is an $A_1^{p,1}$-type gate in $\mathcal{N}$ and φ is an input node to a B_k^1-type gate ψ in $\mathcal{N}$. Let x_k, x_i be the variable input nodes to φ and x_j the variable input node to ψ. Then we can express $f(x_1,\ldots,x_n)$ as $h(f_\psi(x_j, f_\varphi(x_k,x_i)), y_1,\ldots,y_{n-2})$ where $h(z,y_1,\ldots,y_{n-2})$ is some Boolean function of n-1 variables, f_φ and f_ψ are the functions labelling φ and ψ, and $y_1,\ldots,y_{n-2} \in \{x_1,\ldots,x_n\} - \{x_i x_j\}$. Hence $\{f^{x_i,x_j,x_k}\} \subset \{h^{z,x_k}\}$ which has cardinality at most four (since there are only four possible assignments to (z,x_k)), contradicting the assumption that f is $\mathcal{P}_{3,5}$. Q.E.D.

From the restrictions (a) - (d) in Lemma 4.1 we will be able to deduce that $c(f) = |G| \geq \frac{7n-4}{6}$. Note that (a) - (d) only concern the local connectedness structure of $\mathcal{N}$. These local constraints imply linear inequalities relating the number of $A_k^{p,q}$, B_k^p and C_k-type gates in $\mathcal{N}$ which we use to derive our lower bound on c(f).

Let $a_k^{p,q}$, b_k^p and c_k be the number of $A_k^{p,q}$, B_k^p and C_k type of gates in $\mathcal{N}$ respectively. Thus

$$|G| = \sum_{\substack{p\geq q\geq 1,\\ k\geq 0}} a_k^{p,q} + \sum_{\substack{p\geq 1,\\ k\geq 0}} b_k^p + \sum_{k\geq 0} c_k. \qquad (I')$$

Some of the variables in equation (I') can be eliminated. Note that by Lemma 4.1(a), we have $a_k^{1,1} = 0$ for all $k \geq 0$. Also, in any network, variables entering the unique output node must obviously have out-degree exactly one, so $a_0^{p,q} = 0$ for $p \geq q \geq 1$ and $b_0^p = 0$ for $p \geq 2$.

With these variables eliminated, we require a simplifying notation for sums and unions over indices p,q,k. When the range of k is $k \geq 1$, mention of this range will be suppressed, as will be the range of p and q when this range is defined by the pair of conditions $p \geq q \geq 1$, $p \geq 2$.

Thus, under these conventions equation (I') becomes

$$|G| = \Sigma a_k^{p,q} + b_0^1 + \sum_{p\geq 1} b^p{}_k + \sum_{k\geq 0} c_k \qquad (I)$$

From Lemma 4.1, we obtain the following additional inequalities.

Lemma 4.2. $$\Sigma\, a_k^{2,1} \leq 2\cdot\Sigma\, a_k^{2,2} + \sum_{p\geq 3} a_k^{p,2} + \Sigma\, b_k^2. \qquad (II)$$

Proof: An $A_k^{2,1}$-type gate has a unique variable input with out-degree 2. Thus we can define a mapping

$$g: \bigcup A_k^{2,1} \to G$$

by $g(\varphi) = \varphi'$ if the input variable node to φ of out-degree 2 is also an input node to φ'. By Lemma 4.1(d), φ' cannot be an $A_k^{2,1}$-type gate, so φ' is either of type $A_k^{p,2}$, $p \geq 2$, $k \geq 1$, or of type B_k^2, $k \geq 1$. Thus if we let

$K = \bigcup_{p\geq 2} A^{p,2}_k \cup \bigcup B^2_k$, then Range(g) $\subset$ K.

Since each gate has at most two variable input nodes, $|g^{-1}(\varphi')| \leq 2$ for all $\varphi' \in K$. Moreover, if φ' is in $\bigcup_{p\geq 3} A^{p,2}_k \cup \bigcup B^2_k$, then φ' has only one variable input node with out-degree 2, so $|g^{-1}(\varphi')| \leq 1$. Thus $g^{-1}(\varphi') = 2$ only if $\varphi' \in A^{2,2}_k$ for some $k \geq 1$.

The inequality (II) now follows directly: Range(g) $\subset$ K, so $\{g^{-1}(\varphi')\}_{\varphi' \in K}$ gives a partition of Domain(g) $= \bigcup A^{2,1}_k$, and hence

$$\sum a^{2,1}_k = |\bigcup A^{2,1}_k| = |\bigcup_{\varphi' \in K} g^{-1}(\varphi')| = \sum_{\varphi' \in K} |g^{-1}(\varphi')| \leq$$

$$2 \sum a^{2,2}_k + \sum_{p\geq 3} a^{p,2}_k + \sum b^2_k. \qquad \text{Q.E.D.}$$

Lemma 3.3. $$b^1_1 \leq \sum_{k\neq 1} b^1_k + 2\Big(\sum_{p\geq 2} b^p_k + 2 \sum_{k\geq 0} c_k - \sum_{p\geq 2} a^{p,1}_1\Big) \qquad \text{(III)}$$

Proof: A B^1_1-type gate has only one output arc, so it has a unique output node. Thus there is a unique node φ' which is the first non-B^1_1 successor node of a B^1_1-node φ.

Let $\widetilde{B}^1_1 \subset B^1_1$ be the family of B^1_1-type gates which have B^1_k-type gates, $k \neq 1$, as the first non-B^1_1 successor node.

Let $\underset{\sim}{B}^1_1 \subset B^1_1$ be the family of B^1_1-type gates which have B^p_k-type, or C_k-type gates, $p \geq 2$, $k \geq 0$, as the first non-B^1_1 successor node.

Clearly $B^1_1 = \widetilde{B}^1_1 \cup \underset{\sim}{B}^1_1$, and $\widetilde{B}^1_1 \cap \underset{\sim}{B}^1_1 = \emptyset$.

Define

$$\tilde{g} : \tilde{B}_1^1 \to \bigcup_{k\neq 1} B_k^1$$

by $\tilde{g}(\varphi) = \varphi'$ if φ' is the first non-B_1^1 successor node of φ. Note that if $\tilde{g}(\varphi) = \varphi'$, then by Lemma 4.1(b), φ is in fact an input node to φ'. But a B_k^1-type gate has only one input node which is a gate. Hence g is one-to-one, and so

$$|\tilde{B}_1^1| = |\tilde{g}(\tilde{B}_1^1)| \leq |\bigcup_{k\neq 1} B_k^1| = \sum_{k\neq 1} b_k^1.$$

Let $\mathcal{G} = \{\gamma : \gamma$ is an arc from a gate to a B_k^p-type or C_k-type gate, $p \geq 2$, $k \geq 0\}$.

Each B_k^p-type gate has one input arc connected to a gate, while a C_k-type gate has both input arcs connected to gates, so

$$|\mathcal{G}| = \sum_{p\geq 2} b_k^p + 2 \sum_{k\geq 0} c_k.$$

We can define a mapping

$$\underset{\sim}{g} : \underset{\sim}{B}_1^1 \to \mathcal{G}$$

by $\underset{\sim}{g}(\varphi) = \gamma$ if γ is the input arc to the first non-B_1^1 successor node φ' of φ such that γ is in the path from φ to φ'. By Lemma 4.1(b), if $\gamma \in \mathcal{G}$, then $\underset{\sim}{g}^{-1}(\gamma)$ has at most two members, so

$$|\underset{\sim}{B}_1^1| \leq 2\ |g(\underset{\sim}{B}_1^1)|.$$

An $A_1^{p,1}$-type gate φ also has a unique output arc and so a unique output node. By Lemma 4.1(c), the output node of φ cannot be a B_k^1-type gate, so it has to be either a C_k-type or a B_k^q-type gate for $q \geq 2$ and $k \geq 0$. Hence the output arc of φ is in $\mathcal{G}$, and we can define a one-one mapping

$$h : \bigcup_{p>2} A_1^{p,1} \to G$$

by letting $h(\varphi)$ be the output arc of φ. Thus

$$\sum_{p\geq 2} a_1^{p,1} = \Big|\bigcup_{p\geq 2} A_1^{p,1}\Big| = \Big|h\Big(\bigcup_{p\geq 2} A_1^{p,1}\Big)\Big|.$$

Also, if $\gamma \in \underset{\sim}{g}(\underset{\sim}{B}_1^1)$, then γ is the output arc of some B_1^1-type gate, so $\gamma \notin h(\bigcup_{p\geq 2} A_1^{p,1})$. Hence $\underset{\sim}{g}(\underset{\sim}{B}_1^1) \subset G - h(\bigcup_{p\geq 2} A_1^{p,1})$, and so

$$|\underset{\sim}{B}_1^1| \leq 2|\underset{\sim}{g}(\underset{\sim}{B}_1^1)| \leq 2(|G| - |h(\bigcup_{p\geq 2} A_1^{p,1}|)$$

$$= 2(\sum_{p\geq 2} b_k^p + 2\sum_{k\geq 0} c_k - \sum_{p\geq 2} a_1^{p,1}).$$

Thus we conclude that

$$b_1^1 = |B_1^1| = |\widetilde{B}_1^1| + |\underset{\sim}{B}_1^1|$$

$$\leq \sum_{k\neq 1} b_k^1 + 2(\sum_{p\geq 2} b_k^p + 2\sum_{k\geq 0} c_k - \sum_{p\geq 2} a_1^{p,1}) \qquad \text{Q.E.D.}$$

Lemma 3.4. $\sum k a_k^{p,q} - b_0^1 + \sum_{p\geq 1} (k-1) b_k^p + \sum_{k\geq 0}(k-2)c_k = 0.$ (IV)

Proof: Note that total number of output arcs from gates =

$\sum k a_k^{p,q} + \sum_{p\geq 1} k b_k^p + \sum k c_k$. On the other hand, one input node to a B_k^p-type gate is a gate, and both input nodes to C_k-type gate are gates. Since each arc from a gate is both an input arc and an output arc, the total number of output arcs from gates = total number of input arcs that come from gates =

$b_0^1 + \sum_{p\geq 1} b_k^p + 2\sum_{k\geq 0} c_k$. Thus

$$\sum k a_k^{p,q} + \sum_{p\geq 1} k b_k^p + \sum k c_k = b_0^1 + \sum_{p\geq 1} b_k^p + 2\sum_{k\geq 0} c_k, \text{ and (IV) follows.}$$

Q.E.D.

Lemma 3.5. $n = |V|$ = total number of variable nodes in $\mathcal{N}$

$$= \Sigma \left(\frac{1}{p} + \frac{1}{q}\right) a_k^{p,q} + b_0^1 + \sum_{p \geq 1} \frac{1}{p} b_k^p. \qquad \text{(V)}$$

Proof: We assign weights to arcs in $\mathcal{N}$ as follows. Each output arc of a variable node with out-degree p is assigned weight $\frac{1}{p}$, and every other arc is assigned weight 0. Evidently the sum of weights over the output arcs from a variable node is $p \cdot \frac{1}{p} = 1$, and the sum of weights over all arcs is n.

Now, an $A_k^{p,q}$-type gate φ has two variable input nodes, one with out-degree p and the other with out-degree q, and so the sum of weights over input arcs to φ is $\frac{1}{p} + \frac{1}{q}$. Similarly, the sum of weights over input arcs to a B_k^p-type gate is $\frac{1}{p}$, and the sum of weights over input arcs to a C_k-type gate is 0. Hence we also have the sum of weights over all arcs

$$= \sum_{\varphi \in G} \text{(sum of weights over all input arcs to } \varphi)$$

$$= \sum_{A_k^{p,q}} \left(\frac{1}{p} + \frac{1}{q}\right) + \sum_{B_k^p} \frac{1}{p} + \sum_{C_k} 0$$

$$= \Sigma\left(\frac{1}{p} + \frac{1}{q}\right) a_k^{p,q} + b_0^1 + \sum_{p \geq 1} \frac{1}{p} \cdot b_k^p, \text{ and (V) follows.} \qquad \text{Q.E.D.}$$

Finally, since $\mathcal{N}$ has only one node (the output node) with out-degree 0,

$$b_0^1 + c_0 = 1 \qquad \text{(VI)}$$

With inequalities (I) - (VI) we are now ready to find a lower bound for c(f).

Theorem 4.1. If f is $\mathcal{P}^n_{3,5}$, then $c(f) \geq \frac{7n-4}{6}$.

Proof Suppose f is $\mathcal{P}^n_{3,5}$, $\mathcal{N}$ is an optimal network computing f, G is the set of gates in $\mathcal{N}$, and $a^{p,q}_k$, b^p_k and c_k are the number of $A^{p,q}_k$, B^p_k and C_k type gates in $\mathcal{N}$, $p \geq q \geq 1$, $k \geq 0$. Then $c(f) = |G|$, and $\mathcal{N}$ satisfies inequalities (I) - (VI) above. Hence if z^* is the z-value of the optimal solution to the following linear program $(\mathcal{P})$, then $c(f) \geq z^*$.

$$(\mathcal{P}) \quad \text{Minimize} \quad z = \sum a^{p,q}_k + b^1_0 + \sum_{p\geq 1} b^p_k + \sum_{k\geq 0} c_k$$

subject to inequalities (II) - (VI) above and also

$a^{p,q}_k \geq 0$, for $p \geq q \geq 1$, $p \geq 2$, $k \geq 1$.

$b^1_0 \geq 0$,

$b^p_k \geq 0$ for $p \geq 1$, $k \geq 1$, and

$c_k \geq 0$ for $k \geq 0$.

In fact, by the Weak Duality Theorem, if $\bar{v}$ is the v-value of any feasible solution $\bar{y}$ to the dual $(\mathcal{D})$ of $(\mathcal{P})$, then $c(f) \geq z^* \geq \bar{v}$. Hence we only need to find a feasible solution to the dual $(\mathcal{D})$.

The dual $(\mathcal{D})$ of $(\mathcal{P})$ is as follows.

($\mathfrak{D}$) Maximize $v = ny_4 + y_5$

subject to

$$-y_1 - 2y_2 + y_3 + \tfrac{3}{2}y_4 \le 1 \qquad (a_1^{2,1})$$

$$-2y_2 + y_3 + (1 + \tfrac{1}{p})y_4 \le 1, \quad p \ge 3 \qquad (a_1^{p,1})$$

$$-y_1 \qquad ky_3 + \tfrac{3}{2}y_4 \le 1, \quad k \ge 2 \qquad (a_k^{2,1})$$

$$ky_3 + (1 + \tfrac{1}{p})y_4 \le 1, \quad p \ge 3, k \ge 2 \qquad (a_k^{p,1})$$

$$2y_1 \qquad ky_3 + y_4 \le 1, \quad k \ge 1 \qquad (a_k^{2,2})$$

$$y_1 \qquad ky_3 + (\tfrac{1}{p} + \tfrac{1}{2})y_4 \le 1, \quad p \ge 3, k \ge 1 \qquad (a_k^{p,2})$$

$$ky_3 + (\tfrac{1}{p} + \tfrac{1}{q})y_4 \le 1, \quad p \ge q \ge 3, k \ge 1 \qquad (a_k^{p,q})$$

$$y_2 - y_3 + y_4 + y_5 \le 1 \qquad (b_0^1)$$

$$-y_2 + y_4 \le 1 \qquad (b_1^1)$$

$$y_2 + (k-1)y_3 + y_4 \le 1, \quad k \ge 2 \qquad (b_k^1)$$

$$y_1 + 2y_2 + (k-1)y_3 + \tfrac{1}{2}y_4 \le 1, \quad k \ge 1 \qquad (b_k^2)$$

$$2y_2 + (k-1)y_3 + \tfrac{1}{p}y_4 \le 1, \quad p \ge 3, k \ge 1 \qquad (b_k^p)$$

$$4y_2 - 2y_3 + y_5 \leq 1 \qquad (c_0)$$

$$4y_2 + (k-2)y_3 \leq 1, \quad k \geq 1 \qquad (c_k)$$

with $y_1 \geq 0$, $y_2 \geq 0$, and y_3, y_4, y_5 unconstrained in sign.

A feasible solution to ($\mathfrak{D}$) is $\{\bar{y}_1 = 1/12, \bar{y}_2 = 1/6, \bar{y}_3 = -1/3, \bar{y}_4 = 7/6, \bar{y}_5 = -2/3\}$(*), which gives $\bar{v} = n(7/6) + (-2/3) = (7n-4)/6$. Thus $c(f) \geq \bar{v} = (7n-4)/6$. Q.E.D.

In fact, the z*-value of the optimal solution to ($\mathfrak{P}$) is $(7n-4)/6$. If we set all variables except $a_1^{2,1}$, b_0^1, b_1^1, b_1^2 and c_1 to 0, and make all the inequalities into equalities, then ($\mathfrak{P}$) reduces to

$$z^* = a_1^{2,1} + b_0^1 + b_1^1 + b_1^2 + c_1$$

$$\text{subject to} \quad -a_1^{2,1} + b_1^2 = 0,$$

$$-2a_1^{2,1} + b_0^1 - b_1^1 + 2b_1^2 + 4c_1 = 0,$$

$$a_1^{2,1} - b_0^1 - c_1 = 0,$$

(*) The values corresponding to y of the left-hand side of the constraints in ($\mathfrak{D}$) are, in order, $= 1$, $\leq 8/9$, ≤ 1, $\leq 8/9$, ≤ 1, $\leq 13/18$, $\leq 4/9$, $= 1$, $= 1$, ≤ 1, ≤ 1, $\leq 13/18$, $= 2/3$ and ≤ 1.

$$\frac{3}{2}a_1^{2,1} + b_0^1 + b_1^1 + \frac{1}{2}b_1^2 = n,$$

$$b_0^1 = 1.$$

and all variables are non-negative.

Solving the above system of linear equations, we obtain $a_1^{2,1} = b_1^2 = c_1 + 1$, $b_0^1 = 1$, $b_1^1 = 4c_1 + 1$, and $c_1 = \frac{n-4}{6}$, so $z^* = 7c_1 + 4 = \frac{7n-4}{6}$.

The optimal solution to ($\mathcal{P}$) contains some important clues about the kind of gates to use in constructing small networks for $\mathcal{P}_{3,5}$ functions. The "low-cost" $\mathcal{P}_{3,5}$ networks described next were discovered using these clues.

We remark that our lower bound does not take into account the labels of (operations performed by) the gates. Attending to these labels may yield additional constraints on the numbers of different kinds and connections of gates, and may thereby yield a slightly improved lower bound. However, such an analysis appears to be quite tedious, and as the results to follow will show, our bound cannot be much improved.

V. Upper Bounds for $c(\mathcal{P}^n_{3,5})$

In this section we develop a procedure for generating new $\mathcal{P}_{3,5}$ functions and networks from given $\mathcal{P}_{3,5}$ functions and networks. By an argument similar to the one used in Theorem 3.2, we prove in detail that $c(\mathcal{P}^n_{3,5}) \leq \frac{9n-5}{7}$ for infinitely many n; we also indicate how to obtain a slightly better upper bound of $(20n-1)/17$.

Lemma 5.1. Suppose f_1, f_2, g_1 and g_2 are Boolean functions such that $f_1 \neq f_2$, $g_1 \neq 0$, $g_2 \neq 0$ and $(V(f_1) \cup V(f_2)) \cap (V(g_1) \cup V(g_2)) = \emptyset$. Then $f_1 \wedge g_1 \neq f_2 \wedge g_2$.

Proof: Let $F = V(f_1) \cup V(f_2)$. Since $f_1 \neq f_2$, we can, by symmetry, suppose that there is an assignment A of F such that

$$(f_1)^F_A = 0 \text{ and } (f_2)^F_A = 1.$$

Hence $(f_1 \wedge g_1)^F_A = 0$ while $(f_2 \wedge g_2)^F_A = 1 \wedge g_2 = g_2$. But by assumption g_2 is a non-zero function, so $(f_1 \wedge g_1)^F_A \neq (f_2 \wedge g_2)^F_A$ and hence $f_1 \wedge g_1 \neq f_2 \wedge g_2$. Q.E.D.

Lemma 5.2. Suppose f and g are such that $V(f) \cap V(g) = \emptyset$ and f_a^x, g_a^x are non-constant for any variable x and constant a. Then $(f \wedge g)_A^X \neq (\overline{f \wedge g})_B^X$ for any set X of two variables and (not necessarily distinct) assignments A,B of X.

Proof: Let X be a set of two distinct variables and A,B assignments of X. Since $V(f) \cap V(g) = \emptyset$, there are only two possibilities: either (1) $|V(f) \cap X| = 1$ and $|V(g) \cap X| = 1$, or (2) $V(f) \cap X = \emptyset$ or $V(g) \cap X = \emptyset$.

Case 1. $|V(f) \cap X| = 1$ and $|V(g) \cap X| = 1$.

In this case $f_A^X = f^{V(f) \cap X}_a$ for some $a \in \{0,1\}$, so by assumption f_A^X is non-constant. Similarly g_B^X is non-constant. Thus there is an assignment D of V(f) - X such that $f_{A,\ D}^{X,\ V(f)-X} = 0$, and hence $(f \wedge g)_{A,\ D}^{X,\ V(f)-X} = f_{A,\ D}^{X,\ V(f)-X} \wedge g_A^X = 0$. On the other hand, $(\overline{f \wedge g})_{B\ \ D}^{X,V(f)-X} = \overline{f}_{B\ \ D}^{X,V(f)-X} \vee \overline{g}_B^X \neq 0$ because $\overline{f}_{B\ \ D}^{X,V(f)-X}$ is a constant while $\overline{g}_B^X$ is non-constant. Hence we conclude that $(f \wedge g)_A^X \neq (\overline{f \wedge g})_B^X$.

Case 2. $V(f) \cap X = \emptyset$ or $V(g) \cap X = \emptyset$.

By symmetry we can suppose $X \cap V(f) = \emptyset$. Since f_a^x is non-constant, neither is f itself, so there is an assignment D of V(f) such that $f_D^{V(f)} = 0$. But then $(f \wedge g)_{A\ \ D}^{X,V(f)} = f_D^{V(f)} \wedge g_A^X = 0$, while $(\overline{f \wedge g})_{B\ \ D}^{X,V(f)} = \overline{f}_D^{V(f)} \vee \overline{g}_B^X = 1$. Hence, we conclude that $(f \wedge g)_A^X \neq (\overline{f \wedge g})_B^X$. Q.E.D.

Theorem 5.1: Suppose f and g are $\mathcal{P}_{3,5}$ functions such that $V(f) \cap V(g) = \emptyset$ and f_a^x, g_a^x are non-constant for any variable x and constant a. Suppose further that y is a variable not in $V(f) \cup V(g)$, and $h = y \oplus (f \wedge g)$.

Then h is also $\wp_{3,5}$ and h_a^x is non-constant for any variable x and constant a.

Proof: Let X be a set of three distinct variables in V(h). There are two cases: Either (1) $y \notin X$ or (2) $y \in X$.

Case 1. $y \notin X$.

By Lemma 3.2, it suffices to show that $|\{(f \wedge g)^X\}| \geq 5$. There are two sub-cases:

(a) $X \subset V(f)$ or $X \subset V(g)$, and (b) $X \cap V(f) \neq \emptyset$ and $X \cap V(g) \neq \emptyset$.

Case 1(a). $X \subset V(f)$ or $X \subset V(g)$.

By symmetry we can suppose $X \subset V(f)$. Thus $\{(f \wedge g)^X\} = \{f^X \wedge g\}$. But from the assumption, $|\{f^X\}| \geq 5$, so by Lemma 5.1, $|\{f^X \wedge g\}| \geq 5$.

Case 1(b). $X \cap V(f) \neq \emptyset$ and $X \cap V(g) \neq \emptyset$.

By symmetry we can suppose $|X \cap V(f)| = 2$ and $|X \cap V(g)| = 1$. Let $X \cap V(f) = \{x_1, x_2\}$ and $X \cap V(g) = \{x_3\}$.

f is $\wp_{3,5}$, so by Lemma 2.1, it is $\wp_{2,3}$, and thus $|\{f^{x_1,x_2}\}| \geq 3$. Let A_1, A_2, A_3 be assignments of (x_1, x_2) such that

(i) $f^{x_1,x_2}_{A_1}$, $f^{x_1,x_2}_{A_2}$, $f^{x_1,x_2}_{A_3}$ are distinct, and

(ii) $f^{x_1,x_2}_{A_2} \neq 0$, $f^{x_1,x_2}_{A_3} \neq 0$.

By Lemma 5.1 we have

$$f_{Ap}^{x_1,x_2} \wedge g_a^{x_3} \neq f_{Aq}^{x_1,x_2} \wedge g_b^{x_3} \qquad \text{(I)}$$

for $p, q \in \{1,2,3\}$, $p \neq q$ and $a, b \in \{0,1\}$.

Also, by Lemma 2.1, g is $\mathcal{P}_{1,2}$, so $g_0^{x_3} \neq g_1^{x_3}$. Again by Lemma 5.1, we have

$$f_{A_p}^{x_1,x_2} \wedge g_0^{x_3} \neq f_{A_q}^{x_1,x_2} \wedge g_1^{x_3}$$

for $p, q \in \{2,3\}$.

Hence the following five functions in $\{(f \wedge g)^X\} = \{f^{x_1,x_2} \wedge g^{x_3}\}$ are distinct:

$$f_{A_1}^{x_1,x_2} \wedge g_0^{x_3},\quad f_{A_2}^{x_1,x_2} \wedge g_0^{x_3},\quad f_{A_3}^{x_1,x_2} \wedge g_0^{x_3},\quad f_{A_2}^{x_1,x_2} \wedge g_1^{x_3},\quad f_{A_3}^{x_1,x_2} \wedge g_1^{x_3}.$$

Case 2. $y \in X$.

Let $X = \{y\} \cup Y$ where $Y \subset V(f) \cup V(g)$; then $h^X = \{(f \wedge g)^Y\} \cup \{(\overline{f \wedge g})^Y\}$.

Note that Case 1 in fact shows that $f \wedge g$ is $\mathcal{P}_{3,5}$. Thus by Lemma 2.1, $f \wedge g$ is $\mathcal{P}_{2,3}$. Hence $\{(f \wedge g)^Y\}$ and $\{(\overline{f \wedge g})^Y\}$ each contain at least three distinct functions. Moreover, by Lemma 5.2, $\{(f \wedge g)^Y\} \cap \{(\overline{f \wedge g})^Y\} = \emptyset$. Thus $\{h^X\} = \{(f \wedge g)^Y\} \cup \{(\overline{f \wedge g})^Y\}$ contains at least six distinct functions.

Q.E.D.

Suppose we have a $\mathcal{P}^m_{3,5}$ function $h(x_1,\ldots,x_m)$ with the property that h_a^x is non-constant for any variable x and constant a. Also suppose that

we have a network with p gates and m variable nodes computing h.

Define $h^{(1)} = h(x_1,\ldots,x^1_m)$ and

$$h^{(n+1)} = y_n \oplus (h(x^{n+1}_1,\ldots, x^{n+1}_m) \wedge h^{(n)}), \text{ for } n \geq 1,$$

where x^i_j, y_k are distinct variables (so that $\{x^{n+1}_1,\ldots, x^{n+1}_m\} \cap V(h^{(n)}) = \emptyset$ and $y_n \notin \{x^{(n+1)}_n,\ldots, x^{n+1}_n\} \cup V(h^{(n)})$).

Then by n applications of Theorem 5.1, we have

$$h^{(n+1)} \in \wp^{(m+1)n+m}_{3,5}$$

and $h^{(n+1)}$ is computed by a network with $(p+2)n + p$ gates. If we let $N = (m + 1)n + m$ and $P = (p + 2)\, n + p$, then $P = \frac{(p + 2)N - 2m + p}{m + 1}$. Hence we have the following corollary to Theorem 5.1.

Corollary 5.1. Suppose there exists a $\wp^m_{3,5}$ function computable by a network with p gates. Then for infinitely many $N > 0$, there exists a $\wp^N_{3,5}$ function f with $c(f) \leq \frac{(p + 2)N - 2m + p}{m + 1}$.

The network $\mathfrak{M}$ of Figure 10 computes a $\wp^6_{3,5}$ function with the property that, even with any two variables in $\mathfrak{M}$ set to constants, the network does not compute a constant function. (This claim can easily be checked by hand or by computer. Note that because of symmetry, there are only nine sets of two variables and twelve sets of three variables to consider.)

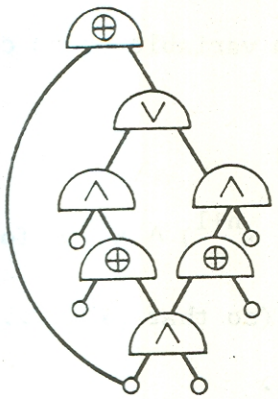


Figure 10. A $\mathcal{P}_{3,5}$ network $\mathfrak{M}$

$\mathfrak{M}$ has 7 gates and 6 variable nodes, thus by Corollary 5.1, for infinitely many N, $c(\mathcal{P}^N_{3,5}) \leq \frac{(7+2)N - 2\cdot 6 + 7}{6+1} = \frac{9N-5}{7}$.

Corollary 5.2. For infinitely many $N > 0$, $c(\mathcal{P}^N_{3,5}) \leq \frac{9N-5}{7}$.

Now consider the network $\mathcal{F}$ with subnetwork $\mathcal{H}$ as shown in Figure 11.

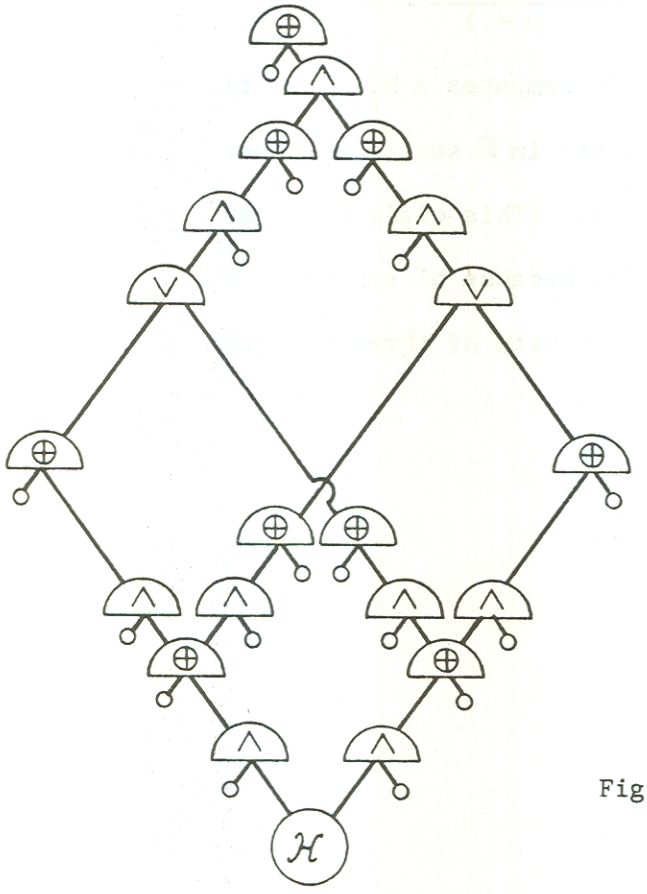


Figure 11. Network $\mathcal{F}$

Note that without considering $\mathcal{H}$, the network $\mathcal{F} - \mathcal{H}$ has 20 gates and 17 variable nodes. Let $V(\mathcal{F})$ denote these 17 variables.

Hsieh [Hs74b] proves the following theorem.

Theorem 5.2. Suppose h is such that

(i) h is $\mathcal{P}_{3,5}$

(ii) h_A^X is non-constant for any set X of two variables and assignment A of X,

and $\mathcal{H}$ is a network computing h. If $V(\mathcal{F}) \cap V(h) = \emptyset$, then the function f computed by the network $\mathcal{F}$ satisfies the same two properties (i) and (ii).

The network $\mathcal{M}$ in Figure 10 satisfies properties (i) and (ii) of Theorem 5.2. Thus by n applications of Theorem 5.2, we have a $\mathcal{P}_{3,5}$ network with $F = 20n + 7$ gates and $N = 17n + 6$ variable nodes. Hence,

$$F = 20\left(\frac{N-6}{17}\right) + 7 = \frac{20N-1}{17}.$$

Corollary 5.3. For infinitely many $N > 0$, $c(\mathcal{P}^N_{3,5}) \leq \frac{20N-1}{17}$.

VI. A Linear Upper Bound for $c(\mathcal{P}^n_{k,2^k})$

In this section we first develop a procedure, similar to that in Section V, for generating new $\mathcal{P}_{k,2^k}$ functions from a given $\mathcal{P}_{k,2^k}$ function. We then present a simple $\mathcal{P}_{k,2^k}$ function due to M. Rabin in $(k + 1)(2k + 3)$ variables with complexity no more than $13(k + 1)(2k + 3)$. We conclude that from the $\mathcal{P}_{k,m}$ property alone only a linear lower bound on combinational complexity can be obtained.

The proofs of the following two lemmas, Theorem 6.1 and Corollary 6.1 are similar to those in Section V and are omitted. (Detailed proofs may be found in [Hs74a.]

Lemma 6.1. If $f \in \mathcal{P}_{k,2^k}$, then for any set $X \subset V(f)$ such that $|X| < k$, and for any assignment A of X, f_A^X is not constant.

Lemma 6.2. If f and g are $\mathcal{P}_{k,2^k}$ functions and $V(f) \cap V(g) = \emptyset$, then $f \wedge g$ is also $\mathcal{P}_{k,2^k}$ and for any set Y of k-1 variables in $V(f) \cup V(g)$ and assignments A,B of Y, $(f \wedge g)_A^Y \neq (\overline{f \wedge g})_B^Y$.

Theorem 6.1. Suppose f and g are $\mathcal{P}_{k,2^k}$ functions with $V(f) \cap V(g) = \emptyset$ and x is a variable not in $V(f) \cup V(g)$. Then $x \oplus (f \wedge g)$ is also $\mathcal{P}_{k,2^k}$.

Similar to Corollary 4.1, we have the following corollary to Theorem 6.1.

Corollary 6.1. Suppose there exists a $\mathcal{P}^m_{k,2^k}$ function computable by a network with p gates. Then for infinitely many $N > 0$, there exists a $\mathcal{P}^N_{k,2^k}$ function f with $c(f) \leq \frac{(p+2)N - 2m + p}{m+1}$.

M. Rabin [Ra74] observed the following simple function to be $\mathcal{P}_{k,2^k}$.

Definition: Let $\mathcal{G}$ be the family of all undirected graphs on $(2n+3)$ vertices $v_1, \ldots, v_{2n+3}$. Define a function f_n on $\mathcal{G}$ as follows:

For $G \in \mathfrak{G}$,

$$f_n(G) = \begin{cases} 1 & \text{if there are two adjacent nodes in G both with degree} \geq n + 3. \\ 0 & \text{otherwise.} \end{cases}$$

A graph on k vertices can be identified with its k×k node-node incidence matrix. Thus $\mathfrak{G}$ is also the family of all $(2n + 3) \times (2n + 3)$ Boolean symmetric matrices with 0's along the main diagonal. Hence f_n can be regarded as a Boolean function of the $(n + 1)(2n + 3)$ Boolean variables $a_{i,j}$ where $1 \leq i < j \leq 2n + 3$.

Lemma 6.3. $f_n \in \mathcal{P}_{n,2^n}^{(n+1)(2n+3)}$.

Proof: Let $P = \{a_{i,j}\}$ $1 \leq i < j \leq 2n + 3$. Note that each assignment, T, of P corresponds to a graph G in $\mathfrak{G}$, and $(f_n)_T^P = f_n(G)$. Henceforth we omit the subscript on f_n.

Suppose N is a set of n variables in P, and A,B are two distinct assignments of N. We need only show that $f_A^N \neq f_B^N$. Thus it is sufficient to find an assignment C of P-N such that $f_{A,\,C}^{N,P-N} = 1$ and $f_{B,\,C}^{N,P-N} = 0$.

Without loss of generality we can suppose $a_{1,2} \in N$, $A(a_{1,2}) = 1$ and $B(a_{1,2}) = 0$. (Recall that for any assignment D and variable x, D(x) denotes the value assigned to x by D.) $N - \{a_{1,2}\}$ has only n - 1 members, so we can certainly find n + 2 indices $k_1,\ldots,k_{n+2} \in \{3,4,\ldots, 2n + 3\}$ such that $a_{1,k_1},\ldots, a_{1,k_{n+2}} \notin N$. Similarly we can find $q_1,\ldots,q_{n+2}$ $\in \{3,4,\ldots, 2n+3\}$ such that $a_{2,q_1},\ldots, a_{2,q_{n+2}} \notin N$.

Let C be the assignment of P-N such that

$$C(a_{1,k_i}) = 1 \quad \text{for } i = 1,\ldots, n+2,$$

$$C(a_{2,q_i}) = 1 \quad \text{for } i = 1,\ldots, n+2,$$

$$C(a_{i,j}) = 0 \quad \text{otherwise.}$$

Let G_1, G_2 be the graphs associated with assignments (A,C) and (B,C), respectively, so that $f^{N,P\text{-}N}_{A,\ C} = f(G_1)$, and $f^{N,\ P\text{-}N}_{B,C} = f(G_2)$. We claim that $f(G_1) = 1$ and $f(G_2) = 0$, which will complete the proof.

Consider $f(G_1)$. $A(a_{1,2}) = 1$, so v_1, v_2 are adjacent in G_1. Moreover, from the definition of C, $V_{k_1},\ldots,V_{k_{n+2}}$ are also adjacent to V_1, so $\deg_{G_1}(v_1) \geq n + 3$; similarly, $\deg_{G_1}(v_2) \geq n + 3$. Hence $f(G_1) = 1$.

Next consider $f(G_2)$. $B(a_{1,2}) = 0$, so v_1, v_2 are nonadjacent in G_2. We assert that for $i \notin \{1,2\}$, $\deg_{G_2}(v_i) \leq n + 2$, from which it follows that $f(G_2) = 0$. Thus let $i \notin \{1,2\}$. The arcs in G_2 that come from assignment C of P-N are incident with either v_1 or v_2, and hence at most two of them can be incident with v_i. However, assignment B of N gives at most n arcs in G_2; thus $\deg_{G_2}(v_i) \leq n + 2$. Q.E.D.

A. Meyer [Me74b] observed that surprisingly f_n has complexity linear in the number of its variables.

Lemma 6.4. $c(f_n) \leq 13(n + 1)(2n + 3)$.

Proof: Note $f_n(P) = \bigvee_{i<j} f_{i,j}(P)$,

where

$$f_{i,j}(P) = \begin{cases} 1 \text{ if } a_{i,j} = 1, \ \sum_{k<i} a_{k,i} + \sum_{k>i} a_{i,k} \geq n+3 \text{ and} \\ \\ \sum_{k<j} a_{k,j} + \sum_{k>j} a_{j,k} \geq n+3, \\ \\ 0 \quad \text{otherwise} \end{cases}$$

For $i = 1,\ldots, 2n+3$, let g_i be such that

$$g_i(a_{i,1},\ldots,\hat{a}_{i,i},\ldots, a_{i,2n+3}) = \begin{cases} 1 \quad \text{if } \sum_{k<i} a_{k,i} + \sum_{k>i} a_{i,k} \geq n+3 \\ \\ 0 \quad \text{otherwise} \end{cases}$$

Then for each pair (i,j) with $i < j$, $f_{i,j} = a_{i,j} \wedge (g_i \wedge g_j)$. Thus if we have networks computing $g_1,\ldots,g_{2n+3}$, then for each (i,j) with $i < j$, we only need two $\wedge$-gates to construct a network for each $f_{i,j}$ and $(n+1)(2n+3) - 1$ $\vee$-gates to combine them. Thus

$$c(f) \leq [(n+1)(2n+3) - 1] + 2\cdot(n+1)(2n+3) + \sum_{i=1}^{2n+3} c(g_i).$$

Now, it is known (for example, see Savage [Sa74]) that for any k, if g is the threshold function defined by

$$g(x_1,\ldots,x_m) = \begin{cases} 1 \quad \text{if } \sum_{i=1}^{m} x_i \geq k, \\ \\ 0 \quad \text{otherwise,} \end{cases}$$

then $c(g) \leq 5m$. Thus for $i = 1,\ldots, 2n+3$, $c(g_i) \leq 5(2n+2)$.

Hence we conclude that

$$c(f) \leq 3(n+1)(2n+3) - 1 + \sum_{i=1}^{2n+3} c(g_i)$$

$$\leq 3(n+1)(2n+3) + 5(2n+2)(2n+3)$$

$$= 13(n+1)(2n+3). \qquad \text{Q.E.D.}$$

The bound in Lemma 6.4 can probably be improved. But the important fact is that we have obtained a $\mathcal{P}_{k,2^k}$ function with linear complexity. Combining Corollary 6.1, Lemma 6.3 and Lemma 6.4, we have the following theorem.

Theorem 6.2: For each $k > 0$, there are infinitely many $n > 0$ with $c(\mathcal{P}^n_{k,2^k}) \leq 13(n+1)$.

It is observed in [Ha73] that the function match on bipartite graphs with 2n vertices:

$$\text{match}(B) = \begin{cases} 1 & \text{if there is a perfect matching in } B \\ 0 & \text{otherwise,} \end{cases}$$

is $\mathcal{P}^{n^2}_{(n-1),2^{n-1}}$, as is the determinant function (modulo 2) of an $n \times n$ Boolean matrix [Sa74]. All known networks computing match or the determinant use at least $O(n^3)$ gates, and we conjecture that their combinational complexity is not linear in the number of variables. Theorem 6.2 reveals that other properties of these functions have to be considered in order to prove a nonlinear lower bound on c (match) or c (determinant).

References

[Eh72] A. Ehrenfeucht, "Practical decidability," Report # CU-CS-OD8-72, Department of Computer Science, University of Colorado, Boulder, Colorado, (1972), 14 pp.

[Ha73] L.H. Harper, with the collaboration of J.E. Savage, "Complexity made simple," to appear in Proceedings of the International Symposium on Combinatorial Theory, Rome, Sept. 2-15, 1973.

[HS72] L.H. Harper and J.E. Savage, "The Complexity of the marriage problem," Advances in Math. 9, 3 (1972), 299-312.

[Hs74a] W.N. Hsieh, "Intersection theorems for systems of finite vector spaces and other combinatorial results," Ph.D. Thesis, Department of Mathematics, M.I.T., Cambridge, Mass., 1974, 51-82.

[Hs74b] W.N. Hsieh, "An Upper Bound for the Complexity of $\mathcal{P}_{3,5}$ Functions," M.I.T. Project MAC Technical Report, Cambridge, Mass. (1974), to appear.

[La70] L.S. Lasdon, Optimization Theory for Large Systems, Macmillan Co., N.Y., 1970, 46-47.

[Lu58] O.B. Lupanov, "On the synthesis of contact networks," Dokl. Akad. Nauk, SSSR, 119, 1 (1958), 23-26.

[Me74a] A.R. Meyer, Lecture notes for course 6.853, Department of Electrical Engineering, M.I.T., Cambridge, Mass., (1974).

[Me74b] A.R. Meyer, Private communication, (1974).

[Ne66] E.I. Nečiporuk, "A Boolean function," Soviet Math. Dokl., 7, 4 (1966), 999-1000.

[Ra74] M. Rabin, Private communication, (1974).

[Sa72] J.E. Savage, "Computational work and time on finite machines," J. ACM 19, 4 (1972), 660-674.

[Sa74] J.E. Savage, "Combinational complexity of functions," chapter II of The Complexity of Computing (1974), to appear.

[Sc74] C.P. Schnorr, "Zwei lineare untere Schranken fur die Komplexität Boolescher Funktionen," (1974), to appear.

[Sh49] C.E. Shannon, "The Synthesis of two-terminal switching circuits," Bell Systems Tech. J. 28, 1 (1949), 59-98.

[St74] L.J. Stockmeyer, "The Complexity of decision problems in automata theory and logic," M.I.T. Project MAC Technical Report 133, Cambridge, Mass. (1974).

Acknowledgement

The authors gratefully acknowledge Albert R. Meyer for suggesting a large number of technical and stylistic improvements to this paper. We also are grateful to Vaughan R. Pratt and Michael J. Fischer for analyzing and revising several examples of $\mathcal{P}_{3,5}$ functions, and for carrying out computer verifications of these examples.