PROCEEDINGS OF THE
AMERICAN MATHEMATICAL SOCIETY
Volume 64, Number 2, June 1977

AN $n \log n$ LOWER BOUND ON SYNCHRONOUS COMBINATIONAL COMPLEXITY[1]

L. H. HARPER

ABSTRACT. Synchronous combinational machines are combinational machines such that the length of all paths from inputs to a logic element are the same. In this paper is is shown that any Boolean function of n variables satisfying certain subfunction conditions (which are satisfied by "almost all" such functions) must have synchronous combinational complexity at least $n \log n$.

0. **Introduction.** In this paper is is shown that if a Boolean function of n variables satisfies certain subfunction conditions (which are satisfied by "almost all" such functions), then its synchronous combinational complexity must be at least $n \log n$.[2] This bound appears to be unique at this time in being nonlinear and applying to a large class of single-valued functions. The restriction on the definition of combinational complexity, i.e., that the machines be synchronous, is very natural from an engineering point of view, and seems to change the general theory of combinational complexity very little. For background on combinational complexity the reader is referred to a survey article by the present author and J. E. Savage entitled *Complexity made simple*.

1. **Definitions.** *A synchronous combinational machine* (s.c.m.), $\mathfrak{M}$, is comprised of

(i) A (finite) directed acyclic graph, G. Each vertex of G will have in-degree at most two, with at most n of them, the set V (for variable), having in-degree zero. The out-degree will be unrestricted, but the set of those of out-degree zero will be contained in the set P (for output), $1 \leqslant |P| = p$. G is called the *graph of* M.

(ii) A one-to-one function φ: $V \rightarrow \{x_1, \ldots, x_n\}$ which assigns to each vertex v in V a Boolean variable $x_i = \varphi(v)$.

(iii) A function κ: $G - V \rightarrow \{f_1, f_2, f_3, f_4\}$ which assigns to each $x \in G - V$ one of the basic Boolean functions

Received by the editors November 18, 1975 and, in revised form, March 7, 1976.
AMS (MOS) subject classifications (1970). Primary 68C05.
Key words and phrases. Boolean functions, combinational complexity, probabilistic set theory.

[1] Research supported by NSF Grant GJ42907.

[2] All logarithms in this paper are taken to the base 2.

$$f_1(x) = x, \qquad f_3(x,y) = x \wedge y,$$
$$f_2(x) = \bar{x}, \qquad f_4(x,y) = x \vee y.$$

κ must assign f_1 and f_2 to vertices of in-degree one and f_3 and f_4 to vertices of in-degree two.

(iv) A rank function r: $G \to Z^+$ such that

(a) $r(v) = 0$ for all $v \in V$.

(b) If there is an edge directed from x to y, then $r(y) = r(x) + 1$.

Note that $\mathfrak{M}$ determines a p-valued Boolean function of the variables $x_1, \ldots, x_n$. In fact the computation of $F(x_1, \ldots, x_n)$ may proceed rank by rank starting at the variable inputs (rank zero) and then progressing to all elements of rank one, then rank two, etc. We may think of signals propagating from the inputs through the machine and condition (iv) insures that if a vertex has in-degree two, then the signals will arrive simultaneously. For this reason the machine is called "synchronous".

For any single-valued Boolean function (hereafter just called "Boolean function" or "function"), f of n variables, the disjunctive normal form

$$f(x_1, \ldots, x_n) = \bigvee x_1^{a_1} \wedge \cdots \wedge x_n^{a_n}, \qquad f(a_1, \ldots, a_n) = 1,$$

where

$$x^a = \begin{cases} x & \text{if } a = 1, \\ \bar{x} & \text{if } a = 0, \end{cases}$$

shows that every Boolean function is computed by some synchronous combinational machine. We define *the cost of a machine* as $|G - V|$, the number of noninput vertices in the machine. We may then define the *synchronous combinational complexity* of a Boolean function f, denoted $C^s(f)$, as the minimum cost of any synchronous combinational machine computing f. Furthermore, if $\mathcal{P}$ is any class of Boolean functions of n variables, then $C^s(\mathcal{P})$, the complexity of the class, is the minimum complexity of any member of the class.

The classes of Boolean functions of n variables for whose complexity we shall obtain lower bounds, are determined by subfunction conditions: Given a p-valued Boolean function, F, integers $1 \leqslant i_1 < i_2 < \cdots < i_m \leqslant n$, and an m-tuple of zeros and ones, $a_1, \ldots, a_m$, we define $F_{a_1 a_2 \cdots a_m}^{x_{i_1} \cdots x_{i_m}}$ to be the function of $n - m$ variables obtained from F (a function of n variables) by setting x_{i_1} to a_1, x_{i_2} to a_2, etc. Then we define $\mathcal{P}_{m,k}$ to be the class of single-valued Boolean functions determined as follows: F is in $P_{m,k}$ if and only if for each choice of $(i_1, \ldots, i_m)$, at least k of the 2^m possible functions $F_{a_1 \cdots a_m}^{x_{i_1} \cdots x_{i_m}}$ are distinct.

In the article by Harper (1975) it is shown that $\mathcal{P}_{m,k}$ is contained in $\mathcal{P}_{m-1,[k/2]}$ and in $\mathcal{P}_{m+1,[\sqrt{k}]}$. Obviously, $\mathcal{P}_{m,k} = \varnothing$ for $k > 2^m$ or $k > 2^{2^{n-m}}$. Now, $k \leqslant \min\{2^m, 2^{2^{n-m}}\}$ is not enough to guarantee that $\mathcal{P}_{m,k}$ is nonempty, but J. Spencer has shown that if $m < m_0(n) - \log\log n$, then $\mathcal{P}_{m,2^m} \neq \varnothing$ and, in fact, contains "almost all" Boolean functions of n variables ($m_0(n)$ is the unique real solution of the equation $2^m = 2^{2^{n-m}}$ and $n - m_0(n) \simeq \log n$).[3] Similarly, he has shown that if $m \geqslant m_0 + 1$ then $\mathcal{P}_m, 2^{2^{n-m}} \neq \varnothing$ and $\mathcal{P}_{m,k} \neq \varnothing$ for $k \lesssim (1 - 1/e)2^m$ and $m \leqslant m_0 + 1$: These relations make it clear that $C^s(\mathcal{P}_{m,k})$ will be maximal for $n - m \simeq \log n$ and $k = 2^m$. In the sequel we choose $m = [n - \log n - \log\log n]$ and $k = 2^m$.

2. **Main results.** Suppose that $\mathcal{M}$ is a synchronous combinational machine computing a Boolean function. Then

$$C^s(f) = \sum_l C_l$$

where C_l is the number of vertices of rank l in the graph of $\mathcal{M}$. If we can show (as we shall) that $f \in \mathcal{P}_{m,k}$, $n - m \simeq \log_2 n$, $k = 2^m$ implies that $C_l \simeq n$ for $1 \leqslant l \lesssim \log n$, then we have that $C^s(f) \gtrsim n \log n$.

LEMMA 1. *Let $\mathcal{M}$ be a s.c.m. computing $f \in \mathcal{P}_{m,k}$. Then for all l and for all $M \subseteq V$, $|M| = m$.*

$$\sum_{r(x)=l} \min\{i(x), 2^{j(x)}\} \geqslant \log k$$

where $i(x) = |M \cap \{v \in V\colon v \text{⧀} x\}|$ and $j(x) = |V - M \cap \{v \in V\colon v \text{⧀} x\}|$.[4]

PROOF. Let ${}_lF$ be the vector-valued Boolean function computed by the machine $\mathcal{M}$ when "decapitated" at rank l. The component functions of ${}_lF$ are computed at the vertices of rank l and if $x \in \mathcal{M}$, $r(x) = l$, then the corresponding component is computed by the submachine whose graph is $\{y \in \mathcal{M}\colon y \text{⧀} x\}$. Also let ${}_lf$ be the function computed by the submachine of $\mathcal{M}$ whose vertices are of rank equal to or greater than l. Obviously $f = {}_lf \circ {}_lF$, and for all $M = \{i_1, \ldots, i_m\}$ and $a_1, \ldots, a_m$,

$$f^{x_{i_1}, \ldots, x_{i_m}}_{a_1, \ldots, a_m} = {}_lf \circ {}_lF^{x_{i_1}, \ldots, x_{i_m}}_{a_1, \ldots, a_m}.$$

Thus

$$|\{{}_lF^{x_{i_1}, \ldots, x_{i_m}}_{a_1, \ldots, a_m}\}| \geqslant k.$$

It is easily seen then that

$$|\{{}_lF^{x_{i_1}, \ldots, x_{i_m}}_{a_1, \ldots, a_m}\}| \leqslant \prod_{r(x)=l} \min\{2^{i(x)}, 2^{2^{j(x)}}\},$$

and so

$$\prod_{r(x)=l} \min\{2^{i(x)}, 2^{2^{j(x)}}\} \geqslant k.$$

[3] By $a \simeq b$ we mean that a is asymptotically equal to b or $a(n)/b(n) \to 1$ as $n \to \infty$. By $a \lesssim b$ we mean $a < b$ for n sufficiently large, or $a \simeq b$.

[4] An acyclic directed graph defines a partial order on its vertices by transitive closure. The "⧀" signs in the definitions of i and j refer to this partial order.

The lemma follows then by taking the logarithm of both sides.

LEMMA 2. *Let $\kappa(n) \to \infty$ as $n \to \infty$. If $\mathcal{F}$ is a family of subsets of a finite set V, $|V| = n$, $|\mathcal{F}| \leqslant n$ and $|E| \leqslant n/\kappa(n)\log n$ for all $E \in \mathcal{F}$; then there exists $M' \subseteq V$, $|M'| = \log n$, such that $\sum_{E\in\mathcal{F}} 2^{|M'\cap E|} \lesssim |\mathcal{F}| + n/\kappa(n)$.*

PROOF. We construct a sequence of sets, $M'_0, M'_1, M'_2, \ldots, M'_{\log n} \subseteq V$ inductively; $M_0 = \varnothing$ and $M'_{m+1} = M'_m \cup \{x_{m+1}\}$, $x_{m+1} \notin M'_m$ for $m > 0$, on the basis of the heuristic that, given M'_m, x_{m+1} should minimize the difference

$$\Delta(x_{m+1}) = \sum_{E\in\mathcal{F}} 2^{|M'_{m+1}\cap E|} - \sum_{E\in\mathcal{F}} 2^{|M'_m\cap E|} = \sum_{\substack{E\in\mathcal{F}\\ x_{m+1}\in E}} 2^{|M'_m\cap E|}.$$

Letting $u_m = \sum_{E\in\mathcal{F}} 2^{|M'_m\cap E|} - |\mathcal{F}|$, we are to show $u_{\log n} \lesssim n/\kappa(n)$. Now, $u_0 = 0$ for $m = 1, 2, \ldots$ and

$$\begin{aligned} u_{m+1} &= \sum_{E\in\mathcal{F}} 2^{|M'_{m+1}\cap E|} - |\mathcal{F}| = \sum_{E\in\mathcal{F}} 2^{|M'_m\cap E|} + \Delta(x_{m+1}) - |\mathcal{F}| \\ &= u_m + \Delta(x_{m+1}). \end{aligned}$$

Also, by the heuristic,

$$\begin{aligned} \Delta(x_{m+1}) &\leqslant \bar{\Delta} = \frac{1}{n-m}\sum_{x\in V-M'_m} \Delta(x) \\ &= \frac{1}{n-m}\sum_{x\in V-M'_m}\sum_{\substack{E\in\mathcal{F}\\ x\in E}} 2^{|M'_m\cap E|} \leqslant \frac{1}{n-m}\sum_{E\in\mathcal{F}} 2^{|M'_m\cap E|}|E| \\ &\leqslant \frac{n}{\log n(n-\log n)\kappa(n)}(u_m + |\mathcal{F}|) \quad \text{for } m \leqslant \log n. \end{aligned}$$

Thus

$$u_{m+1} \leqslant 1 + \frac{n}{\log n(n-\log n)\kappa(n)}u_m + \frac{n^2}{\log n(n-\log n)\kappa(n)}.$$

If we define u'_m by $u'_0 = 0$ and

$$u'_{m+1} = \left(1 + \frac{n}{\log n(n-\log n)\kappa(n)}\right)u'_m + \frac{n^2}{\log n(n-\log n)\kappa(n)},$$

we have $u_m \leqslant u'_m$, and solving this linear first-order difference equation we have

$$u'_m = n\left(1 + \frac{n}{\log n(n-\log n)\kappa(n)}\right)^m - n.$$

Therefore

$$u_{\log n} \lesssim n\left[\left(1+\frac{1}{\log n\,\kappa(n)}\right)^{\log n} - 1\right]$$

$$\simeq n(1/e^{\kappa(n)} - 1) \lesssim n/\kappa(n)$$

and we are done.

THEOREM. *$C^s(\mathcal{P}_{m,k}) \gtrsim n\log n$ for $m = [n - \log n - \log\log n]$ and $k = 2^m$.*

PROOF. Let $\kappa(n)$ be any function on the nonnegative integers such that $\kappa(n) \to \infty$ as $n \to \infty$. Let $\mathcal{M}$ be a s.c.m. computing f, and suppose that $1 \leqslant l \leqslant \log n - \log\log n - \log\kappa(n)$. According to Lemma 1, for all $M \subseteq V$, $|M| = n - \log n$, $\sum_{r(x)=l} \min\{i(x), 2^{j(x)}\} \geqslant \log k$ where $i(x) = |M \cap \{v \in V: v \leqslant x\}|$ and $j(x) = |M' \cap \{v \in V: v \leqslant x\}$ with $M' = V - M$.

In particular,

$$\sum_{r(x)=l} 2^{|M' \cap E(x)|} \geqslant [n - \log n - \log\log n]$$

for all $M' \subseteq V$, $|M'| = \log n$ where $E(x) = \{v \in V: v \leqslant x\}$.

Note. $|E(x)| \leqslant 2^l \leqslant n/\kappa(n)\log n$. But by Lemma 2 there exists some $M' \subseteq V$ such that

$$\sum_{r(x)=l} 2^{|M' \cap E(x)|} \lesssim C_l + n/\kappa(n),$$

so that

$$C_l \gtrsim n - \log n - \log\log n - n/\kappa(n)$$

$$\gtrsim n \quad \text{for } 1 \leqslant l \lesssim \log n$$

and we are done.

3. **Comments and conclusions.** The theorem above may be immediately extended by letting $m = n - n^c$, for $0 < c < 1$. Then by the same argument we have that

$$C_l \gtrsim n - n^c - n/\kappa(n) \quad \text{for } 1 \leqslant l \lesssim (1-c)\log n - \log\kappa(n)$$

so that $C^s(f) \gtrsim (1-c)n\log n$.

At the present time, constructive examples of Boolean functions in class $\mathcal{P}_{m,k}$ for k such that $\log k \simeq m$ and m such that $n - m \simeq \log n$ (or even n^c, for $0 < c < 1$) are not known to the author. The extremal examples now known are a sequence of functions constructed by Meyer and Paterson for which $m = n/\log n$ and $k = 2^m$, and another sequence due to Breitbart which has $m = n/2$ and $k = 2^{n/4} = 2^{m/2} = (2^m)^{1/2}$. Hopefully, this gap will be filled in the near future.

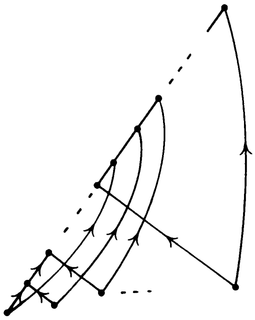
The assumption that our combinational machines be synchronous, though natural from an engineering point of view, was actually dictated by mathematical circumstances: Lemma 1 holds for ordinary (asynchronous) combinational machines, where the role of $\{y \in \mathcal{M}\colon r(y) \leqslant l\}$ is played by any "lower set," L, in the partial order $\circledS$, and the role of $\{y \in \mathcal{M} | r(y) = l\}$ is played by the set of elements L^* in L having outputs connected to members of $\mathcal{M} - L$. Then for all $M \subseteq V$, $|M| = m$, we have that

$$\sum_{x \in L^*} \min\{2^{i(x)}, 2^{2^{j(x)}}\} \geqslant k.$$

This condition extends the conditions of Harper, Hsieh and Savage which gave the lower bounds

$$C(\mathcal{P}_{2,3}) \geqslant n \quad \text{and} \quad C(\mathcal{P}_{3,5}) \geqslant 7/6n - 2/3.$$

However, the machine given schematically by



satisfies this extended condition for $m = n - \log n$, and $k = 2^m$. Thus the best lower bound possible for $C(\mathcal{P}_{m,k})$ by this approach would be $2n$. Of course, only the graph of $\mathcal{M}$ is involved here, so this would not eliminate the possibility that stronger lower bounds could be obtained by taking into account the basic functions computed at each vertex. These problems are quite subtle, and experience has shown that it pays to make these subtle distinctions.

One of the main reasons for being interested in combinational complexity is that it gives a lower bound on the parameters of computation for more general kinds of machines. J. E. Savage has passed along the information that in order for a random access machine (RAM) of S bits of memory to compute a Boolean function in T cycles we must have $(S \log S)T \gtrsim C^s(f)$; also the Fischer-Pippenger argument may be extended to give $ST \gtrsim C^s(f)$ when a Turing machine with tape of length S computes f in T cycles.

Bibliography

1. C. Berge, *The theory of graphs and its applications*, Collection Univ. Math. II, Dunod, Paris, 1958; English transl., Methuen, London; Wiley, New York, 1962. MR **21** #1608; **24** #A2381.

2. Y. Breitbart (to appear).

3. L. H. Harper and J. E. Savage, *Complexity made simple*, Proc. Rome Internat. Sympos. Combinatorial Theory, August 1973 (to appear).

4. L. H. Harper, W. Hsieh and J. E. Savage, *A class of Boolean functions with linear combinatorial complexity*, Theor. Comput. Sci. 1 (1965), 161–183.

5. L. H. Harper, *A note on some classes of Boolean functions*, Studies in Appl. Math. **14** (1975), 161–164.

6. A. Meyer and M. Paterson (to appear).

7. J. E. Savage, *Computational work and time on finite machines*, J. Assoc. Comput. Mach. **19** (1972), 660–674. MR **48** #7661.

8. J. Spencer, Private communication.

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF CALIFORNIA, RIVERSIDE, CALIFORNIA 92502