A NOTE ON
COMPLEX RECURSIVE CHARACTERISTIC FUNCTIONS

by

E. M. McCreight

Carnegie-Mellon University
Pittsburgh, Pennsylvania
May, 1969

This work was supported by the Advanced Research Projects Agency of the Office of the Secretary of Defense (F 44620-67-C-0058) and is monitored by the Air Force Office of Scientific Research. This document has been approved for public release and sale; its distribution is unlimited.

This work was also supported by the Fannie and John Hertz Foundation.

ABSTRACT

There are 0-1 valued functions c(x) such that any program with index i computing c(x) takes a large number of steps except at $\log_2 i$ values of x.

A NOTE ON COMPLEX RECURSIVE CHARACTERISTIC FUNCTIONS

by

E. M. McCreight

A theorem of Rabin[1] asserts there are arbitrarily difficult to compute 0-1 valued recursive functions. In the notation of Blum[2], we have

Theorem 1. (Rabin). $(\forall g \in \mathcal{R}_1)(\exists \text{0-1 valued } c \in \mathcal{R}_1)(\forall i \in \mathcal{N})$
$[\varphi_i = c \Rightarrow (\overset{\infty}{\forall} x)[\Phi_i(x) > g(x)]]$

Although such theorems are of fundamental importance in the study of computational complexity, their applicability to real world computation can be questioned because of the "almost everywhere" $(\overset{\infty}{\forall} x)$ quantification. Thus, the function c of Theorem 1 takes at least g(x) steps to evaluate at x for almost all x, but the set of exceptional x might well include all values of practical interest.

Closer inspection of Rabin's proof reveals that a stronger assertion can be made. Namely, if $\varphi_i = c$, then program i takes at least g(x) steps on input x for all but 2i values of x. Blum has raised the question whether the value 2i can be significantly decreased, in particular to log i.* We present in this note an essentially affirmative answer to Blum's question:

Theorem 2. $(\forall \in > 0)(\forall g \in \mathcal{R}_1)(\exists k \in \mathcal{N})(\exists \text{0-1 valued } c \in \mathcal{R}_1)(\forall_i \in \mathcal{N})$
$[\varphi_i = c \Rightarrow \Phi_i(x) > g(x)$ for all but $k+(1+\in)\log i$ values of x].

Before proceeding to the proof, let us consider the significance

* All logarithms are base 2.

of the log i bound on the number of exceptional values of x. Assume that the Gödel numbering of partial recursive functions is based on a universal computer which, given the binary representations of integers i and x, prints out the value $\varphi_i(x)$ or diverges when $\varphi_i(x)$ is undefined.

Let f be any 0-1 valued recursive function and let β be the binary representation of some Gödel number of f. Let β_n be the binary sequence $f(0),f(1),\ldots,f(n)$. We can now construct a new program for f of the form $\alpha\gamma\beta\gamma\beta_n$ where γ is some fixed binary sequence which serves to separate α,β,β_n, and where α is the binary encoding of the following instructions, "Given input x, see if x+1 $\leq$ length (the string to the right of the second occurrence of γ). If so, print out the xth digit in the string to the right of the second occurrence of γ. Otherwise, compute and print out f(x) according to the instructions given by the string between the first and second occurrences of γ."

The initial portion $\alpha\gamma\beta\gamma$ of the program is independent of β_n, and so there is a constant k = length$(\alpha\gamma\beta\gamma)$ such that for every n, there is a program for f of length n+k which computes the values of f(x) for $x < n$ in the small amount of time required for "table look-up". If $i < 2^{n+k}$ is the index of this program, we have that $\varphi_i = f$ and $\Phi_i(x) <$ table look-up time of x, for all $x < \log i - k$.

It is a routine exercise to construct a Gödel numbering and time measure in which the preceding argument can be carried out formally. In this numbering it is possible to find, for any 0-1 valued recursive functions f, infinitely many indices i for f which compute f rapidly

at log i - k or more different arguments. In short, the bound $(1+\epsilon)\log i$ of Theorem 2 cannot be decreased below log i-k.

Theorem 2 does not bound the size of the values at which program i can run rapidly. Thus, one might initially hope for a result asserting that $\varphi_i = c$ implies $\Phi_i(x) > g(x)$ for all $x > \log i$. Such a result is not possible. By a variation of the argument above, one can devise a Gödel numbering such that for any 0-1 valued recursive function f, and any recursive function h, there are infinitely many indices i for f which compute f(x) rapidly at roughly log i different values of x, and at least one of these values of x exceeds h(i). The proof proceeds by replacing the table β_n by a table of n points on the graph of f. These points are chosen with large abscissas which have relatively small effective descriptions. We leave the details as an amusing exercise for the reader.

Proof of Theorem 2. Let $\epsilon > 0$ and $g \in \mathcal{R}_1$ be given and choose δ so that $0 < \delta < \epsilon$. The required function c is computed in stages, the value c(x) being determined at stage x. At stage 0, each index i has an associated weight equal to $1/(1+i)^{(1+\delta)}$, and is uncancelled. At subsequent stages indices may become cancelled and weights may be increased.

Stage x. Let $A(x) = \{i \leq 2^{2^x} \mid i \text{ is uncancelled, } \Phi_i(x) \leq g(x), \text{ and } \varphi_i(x) = 0\}$.

Let $B(x) = \{i \leq 2^{2^x} \mid i \text{ is uncancelled, } \Phi_i(x) \leq g(x), \text{ and } \varphi_i(x) = 1\}$.

Let $a = \sum_{i \in A(x)} \text{weight}(i)$, and $b = \sum_{i \in B(x)} \text{weight}(i)$.

If $a > b$, define $c(x) = 1$, cancel all $i \in A(x)$, double weight(i) for all $i \in B(x)$, and go to stage x+1.

Otherwise, define $c(x) = 0$, cancel all $i \in B(x)$, double weight(i) for all $i \in A(x)$, and go to stage x+1. END.

The preceding instructions are clearly effective*; hence c is a 0-1 valued recursive function.

Note that $\sum_{i=0}^{\infty} 1/(1+i)^{(1+\delta)} = r$ for some real number $r > 0$, i.e., at stage 0 the cumulative weights of all indices is r. At stage x, the cumulative weights of all uncancelled indices is decreased by $\max\{a,b\}$ and increased by $\min\{a,b\}$. Thus the cumulative weights of all uncancelled indices at any stage remains bounded above by r. It follows that the weight of any index i can, in the course of the above procedure, be doubled at most $\log r + (1+\delta)\log(1+i)$ times (or else weight(i) would exceed r while i was uncancelled).

Suppose $\varphi_i = c$. Then index i is never cancelled (since c differs from φ_i for all cancelled i by construction). For each $x \geq \log\log i$ such that $\Phi_i(x) \leq g(x)$, weight(i) must therefore be doubled at stage x. It follows that $\Phi_i(x) \leq g(x)$ for at most $\log\log i + \log r + (1+\delta)\log(1+i)$ different values of x. But

$$\log\log i + \log r + (1+\delta)\log(1+i) < k + (1+\epsilon)\log i$$

for some constant $k \in \mathbb{N}$ and all i. Q.E.D.

The same proof yields slightly stronger bounds, e.g.,

Corollary. $(\forall \epsilon > 0)(\forall g \in \mathcal{R}_1)(\exists k \in \mathbb{N})(\exists$ 0-1 valued $c \in \mathcal{R}_1)(\forall \varphi_i = c)$
$[\Phi_i(x) > g(x)$ for all but $k + \log i + (1+\epsilon)$ loglog i values of x].

The corollary is proved by setting $\text{weight}(i) = 1/(i+1)\log(i+1)(\text{loglog}(i+1))^2$ since $\sum \text{weight}(i) < \infty$ in this case also. The corollary

---

* There is a technical problem arising from the use of real numbers. Namely, the test "a > b?" may not be effective. This defect can be overcome by defining weight(i) to be a rational approximation to $1/(i+1)^{(1+\delta)}$ at stage 0.

can still be strengthened by using more slowly convergent series, but it remains an open question whether the bound can be decreased to $k + \log i$.

Acknowledgment

The author would like to thank Prof. Albert R. Meyer for help both in sharpening the original proofs and writing this note.

Notes

1. Rabin, M. O., Degree of difficulty of computing a function and a partial ordering of recursive sets. Tech. Rep. No. 2, Hebrew University, Jerusalem, Israel (1960).

2. Blum, M. A machine-independent theory of the complexity of recursive functions. JACM, 14, 2 (April, 1967), 322-336.

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)	2a. REPORT SECURITY CLASSIFICATION
Carnegie-Mellon University Department of Computer Science Pittsburgh, Pennsylvania 15213	UNCLASSIFIED 2b. GROUP

3. REPORT TITLE

A NOTE ON COMPLEX RECURSIVE CHARACTERISTIC FUNCTIONS

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Scientific Interim

5. AUTHOR(S) (First name, middle initial, last name)

E. M. McCreight

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
May 1969	7	2

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
F44620-67-C-0058 b. PROJECT NO. 9718 c. 61 54 501R d. 681304	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT

1. This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
TECH, OTHER	Air Force Office of Scientific Research (SRMA) 1400 Wilson Boulevard Arlington, Virginia 22209

13. ABSTRACT

There are 0-1 valued functions c(x) such that any program with index i computing c(x) takes a large number of steps except at $\log_2 i$ values of x.

DD FORM 1 NOV 68 1473

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT