

REFINEMENT OF HIERARCHIES OF
TIME BOUNDED COMPUTATIONS.

J. Hartmanis

J. E. Hopcroft

Technical Report

No. 68-20

June 1968

Department of Computer Science
Cornell University
Ithaca, New York 14850

REFINEMENT OF HIERARCHIES OF
TIME BOUNDED COMPUTATIONS.*

by

J. Hartmanis

J. E. Hopcroft

ABSTRACT

It is shown that for any "slowly growing" time function $T(n)$ and any $\epsilon > 0$ there exists a computation which can be performed by a multitape Turing machine in time $T(n)\log^{\epsilon}T(n)$ and cannot be performed by any multitape Turing machine in time $T(n)$.

*This research has been supported in part by National Science Foundation Grant GP-6426.

INTRODUCTION

The computational complexity of a sequence has been measured [1] by the time it takes a multitape Turing machine to print out the successive symbols of the sequence. Sequences are placed in complexity classes as follows. If $T(n)$ is a computable, monotone increasing function of positive integers into positive integers and if α is a binary sequence, then we say that α is in complexity class $C_{T(n)}$ if and only if there is a multitape Turing machine M such that M computes the n th symbol of α within $T(n)$ operations.

Hennie and Stearns [2] have shown that for a real-time computable function [4] $T(n)$, a complexity class $C_{T_1(n)}$ is properly contained in $C_{T(n)}$ if

$$\lim_{n \rightarrow \infty} \frac{T_1(n) \log T_1(n)}{T(n)} = 0 .$$

In this paper we show that this result can be considerably improved for slowly growing time bounds. Let $T(n)$ be a real-time computable function which satisfies the inequality

$$T(n) \leq T(n) \log^c T(n)$$

for any $c > 0$ and sufficiently large n . Then we show that for any $\epsilon > 0$ the complexity class $C_{T(n)}$ is properly contained

in $C_{T(n)\log^c T(n)}$. (In all our applications we consider the value of a function to be its integer part.)

REFINEMENT OF TIME-BOUNDED HIERARCHIES

Theorem 1: For any real-time computable function $T(n)$ and for any $\epsilon > 0$ there exists an integer k and a sequence α such that α is in complexity class $C_{T(n)\log^{(k+1)\epsilon} T(n)}$ but not in complexity class $C_{T(n)\log^{k\epsilon} T(n)}$.

Proof: From the Hennie-Stearns result it follows that

$$C_{T(n)} \subset C_{T(n)\log^2 T(n)}. \text{ Clearly for each } k = 0, 1, 2, \dots,$$

$$C_{T(n)\log^{k\epsilon} T(n)} \subset C_{T(n)\log^{(k+1)\epsilon} T(n)}. \text{ Thus for some integer}$$

k the containment is proper, as was to be shown.

This shows that for any real-time computable function $T(n)$ and $\epsilon > 0$ there exists a $U(n) = T(n)\log^{k\epsilon} T(n)$ such that

$$C_{U(n)} \subset C_{U(n)\log^\epsilon U(n)};$$

since

$$\begin{aligned} U(n)\log^\epsilon U(n) &= [T(n)\log^{k\epsilon} T(n)]\log^\epsilon [T(n)\log^{k\epsilon} T(n)] \\ &\geq T(n)\log^{(k+1)\epsilon} T(n) \end{aligned}$$

and therefore by Theorem 1 there is an α in $C_{U(n) \log^\epsilon U(n)}$ but not in $C_{U(n)}$.

Note that, if $T(n)$ is real-time computable and ϵ is a positive rational, then $T(n) \log^\epsilon T(n)$ is again a real-time computable function. Thus we see that the Hennie-Stearns theorem can be considerably strengthened for infinitely many cases and it leads us to conjecture that this stronger result holds for all real-time functions.

Conjecture. For any real-time computable function $T(n)$ and $\epsilon > 0$

$$C_{T(n)} \subset C_{T(n) \log^\epsilon T(n)}$$

The next result shows that the conjecture holds for real-time computable functions satisfying

$$T(n+1) \leq T(n) \log^\epsilon T(n)$$

for $\epsilon > 0$ and n sufficiently large.

Theorem 2: Let $T(n)$ be a real-time computable function such that for some $\epsilon > 0$ and large n $T(n+1) \leq T(n) \log^\epsilon T(n)$.

Then there exists a sequence α in $C_{T(n) \log^\epsilon T(n)}$ not in

$C_{T(n)}$.

Proof: By Theorem 1, for some integer k there exists a sequence $\beta = \beta_1\beta_2\dots$ in $C_{T(n)\log^{(k+1)\epsilon}T(n)}$ not in $C_{T(n)\log^{k\epsilon}T(n)}$.

Let m_n be the smallest integer greater than or equal to $T^{-1}(T(n)\log^{k\epsilon}T(n))$. Consider the sequence $\alpha = \alpha_1\alpha_2\dots$ defined as follows:

- 1) $\alpha_{m_n} = \beta_n$ for $n = 1, 2, \dots$
- 2) $\alpha_i = 0$ if $i \neq m_n$ for any n .

Clearly α is not in $C_{T(n)}$ since for each n , α_{m_n} must be computed in

$$T(m_n) \geq T[T^{-1}(T(n)\log^{k\epsilon}T(n))] = T(n)\log^{k\epsilon}T(n),$$

which contradicts the assumption that β is not in $C_{T(n)\log^{k\epsilon}T(n)}$.

We now show that α can be computed in time $T(n)\log^{\epsilon}T(n)$. To do this we will use a Turing machine with $s+t+1$ tapes where s is the number of tapes needed to compute the characteristic sequence corresponding to $T(n)\log^{\epsilon}T(n)$ and t is the number of tapes needed to compute the sequence β in time $T(n)\log^{k\epsilon}T(n)$. (Note If $T(n)$ is real-time computable, then $T(n)\log^{\epsilon}T(n)$ is real-time computable for any rational $\epsilon > 0$.) The first tape will have two heads. One head will be used to print the symbols of β as they are computed and the other will be used

to read the symbols when they are needed for output. (It is known [1] that we can replace without loss of computation speed a Turing machine with several heads per tape by a Turing machine with one head per tape by increasing the number of tapes.) Tapes 2 through $t+1$ will be used to compute the sequence β . Each time a symbol is computed it is stored on tape 1. One of the heads on tape 1 is used solely for this purpose. Tapes $t+2$ through $s+t+1$ are used to compute the characteristic sequence corresponding to $T(n)\log^c T(n)$. At each time unit for which the characteristic sequence is 0, the Turing machine outputs a 0. Whenever the characteristic sequence is 1, the Turing machine outputs the symbol under the second head on tape 1 and then moves the head one cell right. Note that tape 1 is used solely for the purpose of storing the symbols of β until they are to be outputted. It remains to show that we never attempt to output a symbol before it is computed. Now β_n is computed within time $T(n)\log^{(k+1)c} T(n)$ and is outputted at time $T(m_n)\log^c T(m_n)$. Since $m_n \geq T^{-1}[T(n)\log^{kc} T(n)]$ we know that

$$\begin{aligned} T(m_n)\log^c T(m_n) &\geq T(n)\log^{kc} T(n)\log^c (T(n)\log^{kc} T(n)) \\ &\geq T(n)\log^{(k+1)c} T(n) \end{aligned}$$

for sufficiently large n . Thus β can be computed in time

$T(n) \log^{\epsilon} T(n)$ and we conclude that

$$C_{T(n)} \subset C_{T(n) \log^{\epsilon} T(n)}$$

as was to be shown.

For real-time computable functions which satisfy

$$T(n+1) \leq T(n) \log^{\epsilon} T(n)$$

for all $\epsilon > 0$, Theorem 2 yields that

$$C_{T(n)} \subset C_{T(n) \log^{\epsilon} T(n)}$$

for all $\epsilon > 0$ which is a considerable improvement on the Hennie-Stearns result. Unfortunately, the inequality bounds the growth rate by some $B_1(n)$ such that

$$2^{n(\log n)^{1-\delta}} < B_1(n) < 2^{n \log n}, \quad \delta > 0.$$

The growth rate of functions satisfying

$$T(n+1) \leq T(n) \log T(n)$$

is bounded by

$$B_2(n) = e^{E_1^{-1}(n)}$$

where

$$E_1(n) = \int_{-\infty}^n e^y / y \, dy .$$

Clearly

$$n^n < B_2(n) < n^{n(1+\delta)} , \quad \delta > 0 .$$

For these functions, Theorem 2 yields that

$$C_{T(n)} \subseteq C_{T(n)\log T(n)}$$

which still does not follow from Hennie-Stearns. Observe that

$$2^{n(\log n)^{1-\delta}} < B_1(n) < 2^{n \log n} = n^n < B_2(n) < n^{n(1+\delta)}$$

for any $\delta > 0$, thus the functions

$$k^n , \quad k > 1$$

$$n^k (\log n)^{p/q} , \quad k \geq 1$$

$$n^k (\log \log n)^{p/q} , \quad k \geq 1 , \text{ etc.}$$

satisfy the inequality for any $\epsilon > 0$ and the functions

$$n!$$

$$n^n$$

$$2^{n(\log n)^{1-\delta}} , \quad \delta > 0$$

satisfy the inequality for $\epsilon = 1$.

CONCLUSION

It was shown in this paper that the Hennie-Stearns theorem for time-bounded computations can be uniformly improved for slowly growing time functions and that it can be improved for infinitely many arbitrarily fast growing time functions. Clearly, the interesting problem remains to determine whether

$$C_{T(n)} \subseteq C_{T(n) \log^{\epsilon} T(n)}, \quad \epsilon > 0,$$

for all real-time computable functions (as the first result seems to suggest).

It should be noted that it was known [3] that the Hennie-Stearns result can be improved for some special cases. All the cases in [3] for which proofs are given follow as special cases of Theorem 2, and the techniques used in [3] seem to be limited by the bound $B_2(n)$.

ACKNOWLEDGEMENT

The authors are indebted to R. J. Walker and A. Borodin for helpful discussions on this problem.

REFERENCES

1. J. Hartmanis and R. E Stearns, On the Computational Complexity of Algorithms, Transactions of the American Mathematical Society 117 (May 1965), 285-306.
2. F. C. Hennie and R. E. Stearns, Two Tape Simulation of Multitape Turing Machines, JACM (October 1966), 533-546.
3. S. S. Ruby and P. C. Fischer, Translation Methods and Computational Complexity, Proc. Sixth Ann. Symp. on Switching Circuit Theory and Logical Design, 1965, 173-178.
4. H. Yamada, Real-time Computation and Recursive Functions not Real-time Computable, IRE Transactions EC-11 (1966), 753-760.

