# Purdue University 

## Purdue e-Pubs

# A Software Physics Analysis of Akiyama's Debugging Data 

Yasao Funami

M. H. Halstead

## Report Number:

75-144

Funami, Yasao and Halstead, M. H., "A Software Physics Analysis of Akiyama's Debugging Data" (1975). Department of Computer Science Technical Reports. Paper 93.
https://docs.lib.purdue.edu/cstech/93

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries. Please contact epubs@purdue.edu for additional information.

# A SOFTWARE PHYSICS ANALYSIS OF AKIYAMA'S DEBUGGING DATA 

Yasao Funami<br>and<br>M. H. Halstead

May 1975

CSD-TR 144

A SOFTWARE PHYSICS ANALYSIS OF AKIYAMA'S DEBUGGING DATA

Yasao Funami
M. H. Halstead
F. Akiyama [1] has published a careful study of the number of bugs which occurred in the programming of each of the nine modules of a 100 manmonth software system called SAMPLE. All of his observed data are reproduced in Table 1. In the case of Module MC, 53 bugs were reported before machine runs were obtained, and these have been included.

Table 1. Akiyama's Observations

| Program Module | MA | MB | MC | MD | ME | MF | MG | MH | MX |  |
| :--- | ---: | ---: | ---: | ---: | ---: | ---: | ---: | ---: | ---: | ---: |
| Program Steps (S) | 4032 | 1329 | 5453 | 1674 | 2051 | 2513 | 699 | 3792 | 3412 |  |
| Decisions | (D) | 372 | 215 | 552 | 111 | 315 | 217 | 104 | 233 | 416 |
| Calis | (J) | 283 | 44 | 362 | 130 | 197 | 186 | 32 | 110 | 230 |
| Number of Bugs (B) | 102 | 18 | 146 | 26 | 71 | 37 | 16 | 50 | 80 |  |

In presenting his data, Akiyama reported that the coefficient of correlation between number of bugs and number of program steps was 0.83 , while the correlation between bugs and the sum of decisions plus calls was much higher, at 0.92.

An interesting and quantitative explanation of this result is provided by the theory of software physics [2]. According to that theory, the number of effective mental discriminations, E, require for the implementation of a program is given by:

$$
\begin{equation*}
E=V / L=\left(N \log _{2} \pi\right) /\left(n_{1}^{*} \pi_{2} / \eta_{1} N_{2}\right) \tag{1}
\end{equation*}
$$

where:
$V=$ Program volume.
$L=$ Program level.
$n_{1}^{\frac{1}{*}}=2=$ Unique operators required by a call.
$\eta_{1}=$ Unique operators used in the program.
$\eta_{2}=$ Unique operands used in the program.
$\mathrm{N}_{2}=$ Total usage of operands.
$N=$ Total usage of operands and operators.
$\eta=\eta_{1}+n_{2}$
While Akiyama's data do not include these parameters directly, they do supply observations from which they may be estimated. If we assume that each of the $S$ machine language steps includes one operator and one operand, then:

$$
\begin{equation*}
N_{2}=S \tag{2}
\end{equation*}
$$

and $N=2 S$.
The number of unique operators, ${ }^{7}$, is composed of three classes of operators. The first is the number of distinct operators used from the machine's repertoire of instructions. For large programs, this component may be roughly approximated as an octal hundred. Second is the number of distinct operations provided by functions or subroutines. This component should correspond to item $J$ in Table l. Finally, each transfer to a unique location has been shown by Bulut [3] to contribute directly to $\eta_{1}$. Since the number of transfers implied by item $D$ in Table 1 do not each involve transfer to a unique location, only a fraction, perhaps one third, should contribute to $\eta_{1}$. We then have, roughly:

$$
\begin{equation*}
\eta_{1}=0 / 3+J+64 \tag{4}
\end{equation*}
$$

At this point, we need only an estimate of $\eta_{2}$ to be able to calculate $E$. From the length equation as presented by Halstead and Bayer [4] and independently validated by Bohrer [5].

$$
\begin{equation*}
N=n_{1} \log _{2} n_{1}+n_{2} \log _{2} \eta_{2} \tag{5}
\end{equation*}
$$

It is possible to find $n_{2}$ when $\eta_{1}$ and $N$ are known.
Using equations 1 through 5, the data of Table 1 yield the results shown in Table 2.

Table 2. Software Physics Parameters derived from Table 1.

| Module | MA | MB | MC | MD | ME | MF | MG | MH | MX |
| :--- | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| N | 8064 | 2658 | 10906 | 3348 | 4102 | 5026 | 1398 | 7584 | 6824 |
| $\mathrm{~N}_{2}$ | 4032 | 1329 | 5453 | 1674 | 2051 | 2513 | 699 | 3792 | 3412 |
| $\eta_{\mathrm{I}}$ | 471 | 180 | 610 | 231 | 366 | 322 | 131 | 252 | 433 |
| $\eta_{2}$ | 442 | 176 | 574 | 201 | 138 | 287 | 76 | 603 | 357 |
| E (Millions) | 170.3 | 15.3 | 322.6 | 28.2 | 100.2 | 65.5 | 6.5 | 58.5 | 135.9 |

The correlation coefficient between number of effective mental discriminations, E , and the reported number of bugs, B , is 0.982 , indicating that mos.t of the variation has been explained.

Further, by using the usual figure of 18 mental discriminations per second for fluent, concentrating programmers [6], (Stroud [7] gives the range as 5 to 20 per secorid), and summing the values of $E$, one obtains the total effort of the task as $903 \times 10^{6}$ effective discriminations, or 84 man-months. This figure compares reasonably well with the 100 man-months reported by Akiyama.

## REFERENCES:

[1] Akiyama, F,, "An Example of Software System Debugging" Proc. IFIP Congress, 1971.
[2] Halstead, M. H., "Software Physics Comparison of a Sample Program in DSL ALPHA and COBOL," IBM Research Report RJ1460, October 1974.
[3] Bulut, Necdet, "An Invariant Property of Algorithms," Ph.D. Thesis, Purdue University, August 1973.
[4] Halstead, M. H. and Rudolf Bayer, "Algorithm Dynamics," Proc. ACM National Conference, 1973 (Atlanta).
[5] Bohrer, Robert, "Halstead's Criterion and Statistical Algorithms," Proc. Eighth Annual Computer Science/Statistics Interface Symposium, February. 1975 (Los Angeles).
[6] Halstead, M. H. "Toward a Theoretical Basis for Estimating Programming Effort," Technical Report CSD-TR 143, Purdue University, May 1975.
[7] Stroud, John M. "The Fine Structure of Psychological Time," Annuals of New York Academy of Sciences, 1966, pp. 623-631.

