Human Memory Organization for Computer Programs

A. F. Norcio
U.S. Naval Academy, Annapolis, MD 21402

Stephen M. Kerst
Catholic University of America, Washington, DC 20017

Human memory organization has been shown to be important in the processing of natural language. Evidence is provided in this investigation which suggests that human memory organization is also important in processing programming languages. Subjects were divided into experimental groups which studied programs with or without documentation, and with or without hierarchically indented statements. Subjects studied and recalled five Fortran programs. The pattern of recalled statements at logic segment boundaries was compared to the recalled pattern within segments. In addition, the recalled boundary patterns of the experimental groups were compared to each other. The results indicate that algorithmic logic segments form a cognitive organizational structure in human memory for computer programs. Statement indentation and internal program documentation did not apparently enhance the organizational process or the recall of programming statements.

Introduction

Miller [16] introduced into the psychological literature the term "chunking," which he used to describe the "process of organizing or grouping the input into familiar units or chunks" (p. 93). A good example of this process is the natural phenomenon of recoding letters into words, words into sentences, sentences into paragraphs, and so on. In programming, DO loops, input statements, and subroutines provide examples of natural groupings. The term chunking refers specifically to the recoding process of short term memory. The term organization has been used in the literature to denote the corresponding process in long term memory. Klatzky [14] notes that organization and chunking are "fundamentally the same process" (p.190). The distinction between them is the research setting (long or short term memory) used to examine this cognitive process.

Received March 11, 1982; revised July 16, 1982; accepted August 9, 1982

The present study investigated experimentally the relationship between the fundamental process of memory organization and algorithmic logic segments of computer programs. Memory organization theory claims that logically related information units are recoded into distinctive groupings. Algorithmic segments of a computer program constitute well defined units which would seem to provide a basis for organization in memory. If subjects correctly recall statements in groups which correspond to algorithmic logic segments, it would suggest that they have cognitively organized program statements by logical segments.

Human Memory Organization and Information Processing

Miller [16] underlined the importance of human memory organization when he described it as "lifeblood of the thought processes" (p. 95). Insightful research into the human memory organization process has been provided by Neal Johnson [8–13]. Johnson's evidence shows that subjects use the phrase structure segments of a sentence as chunks in memory tasks; i.e., more recall errors are found at phrase boundaries than within phrases. Ammon [1] and Suci [26] have also shown that sentence phrase boundaries tended to produce greater errors in recall. In addition, Fodor and Bever [6] have shown the significance of syntactic boundaries of linguistic segments at the deep structural level.

Memory organization effects are not restricted to the recall of single sentences. Crohers [4] and Frederiksen [7] have shown that memory organization is important in the cognitive processing of complete prose passages. Egan and Schwartz [5] have demonstrated that chunking is involved in the recall of schematic circuit diagrams. Chase and Simon [2] have reported that chunking is a major factor in the recall of chess pieces in complex positions. It is noteworthy that expert chess players organized the information into larger chunks and recalled more positions than novices. These and other results suggest that organized memory for logical and strategic information is a crucial component in the expert's skill. It seems likely that memory organization plays a role in complex cognitive tasks, such as computer programming.

Human Memory Organization and Computer Programming

Some suggestive evidence for the role of memory in programming was found by Love [15], who examined the relationship between individual differences in programming performance and short term memory processing ability. He observed that individuals with high scores on short term memory tests made fewer logical programming errors. Independent variables which several researchers have investigated are hierarchical indentation of programming statements, internal program documentation, and mnemonic variable names. Love [15], Shneiderman, and McKay [21], and Weissman [29,30] have reported that students preferred statement indentation, but they did not find significant effects on recall. It is not clear whether the nonsignificant results show that indentation is generally ineffective or whether the specific indentation used was somehow inappropriate for novice programmers. Shneiderman [19] and Shneiderman and Mayer [20] have also used internal documentation and mnemonic variable names as independent variables in measuring program recall. They found that subjects who were given initial program documentation were able to recall more program statements correctly than subjects given interspersed documentation. Other psychological research related to computer programming has dealt with specific constructs such as IF statements [23-25] or algorithmic problem solving [17], but human memory organization for computer programs has not been examined.

The purpose of the present study was threefold. First, it was desired to determine how human memory organization for a computer program is related to the algorithmic organization of the program itself. Second, the effects of the placement of documentation and indentation on memory organization and recall were examined. In particular, such effects were studied when placement was determined by the boundaries of algorithmic segments which were identified by subjects like those who attempted to recall the programs. If such psychologically appropriate placement enhanced memory organization, one might expect a consequent improvement in recall. Third, the relationship between the degree of memory organization and recall in conventional programs and those with indentation and documentation was investigated.

The recall patterns of statement transitions were used to determine the relationship between memory organization and algorithmic logic segments. In recalling a computer program, statement to statement transitions can occur in four mutually exclusing ways. A correct statement can be followed by a correct statement (C/C); or it can be followed by an incorrect statement (C/I). Conversely, an incorrect statement can be followed by either a correct (I/C) or an incorrect statement (I/I). The statement transition pattern can be defined as the observed pattern of transitions in a subject's written recall of a program. A boundary transition can be defined as a transition in which one statement is the last statement in a particular logic segment and the succeeding statement is the first statement in the following segment. Memory organization by algorithmic logic segments would be suggested during recall by significantly higher frequencies of C/I or I/C transitions at segment boundaries than at nonsegment boundaries.

Method

Subjects

The subjects, who were in their first or second year of college, had completed their first programming course. They had no prior computer experience before the course. Ten subjects were randomly selected and randomly assigned to a panel which defined the algorithmic logic segments in each program. Twenty subjects were randomly assigned to two pilot studies of ten subjects each which determined the appropriate presentation and recall times. Sixty subjects were assigned to six experimental conditions of ten subjects each. The experimental conditions constituted a 2×3 crossing of two levels of the indentation factor (present or absent) with three levels of the documentation factor (none, initial, interspersed).

Materials

Five unstructured Fortran programs which were appropriate in complexity for novice programmers were selected.* The length of each program was fifteen to thirty-three programming statements. The programs did not presuppose any specific mathematical or logical knowledge. Program One converted Roman numerals to their corresponding Arabic numbers. Program Two calculated change with a minimum number of denominations of currency and coins based upon the purchase price and cash. Given the position of a knight on a chess board, Program Three determined all the possible moves the piece can legally make. Program Four determined the median of a given array of numbers. Program Five calculated the sum of squares of an array of numbers. (See Fig. 1.)

Procedure

In order to empirically define logic segment boundaries in each program, each panel member was shown the unindented programs in individual sessions. The panel members were instructed to divide each program into logic segments. The most frequently chosen boundaries defined the algorithmic logic segments in each program. The

*Copies of the programs can be obtained by contacting Dr. A. F. Norcio.

```
C THIS PROGRAM COMPUTES THE SUM OF THE
C SQUARED DEVIATIONS FROM THE MEAN FOR A
C GIVEN ARRAY OF NUMBERS.  THE PROGRAM USES
C THE COMPUTATIONAL VERSION OF THE SUM OF SQUARES
C FORMULA
C
        DIMENSION X(100)
1       READ(5,10) N
10      FORMAT(I3)
        READ(5,20) (X(I),I=1,N)
20      FORMAT(F5.2)
            SMX=0

                DO 30 I=1,N
                    SMX2=SMX2+X(I)**2
                    SMX=SMX+X(I)

            SS=SMX2-(SMX**2/N)
        WRITE(6,60) SS

        END
```

FIG. 1. Sample program with initial documentation and identification.

panel's definitions also determined the location of the interspersed documentation and the indentation hierarchy.

A pilot study of twenty students was conducted to determine appropriate times for the presentation and recall of the programs. In the pilot study, only the unindented undocumented programs were used. Ten pilot subjects were given seven presentation minutes and seven minutes for recall; the second ten pilot subjects were given fifteen minutes for presentation and fifteen minutes for recall. The pilot subjects were tested individually. From the pilot study five minutes for presentation and five minutes for recall were judged appropriate since no pilot subject spent longer than five minutes in either task.

Each experimental subject was given five minutes to examine the first program. Following this, each subject was given five minutes for written recall of the program. This process was repeated until each subject had examined and recalled all programs. The programs were presented to the experimental subjects in random order to counterbalance possible carryover or order effects. Forty subjects were run in two classes of thirty and ten students each, and twenty students performed the experiment on a scheduled individual basis.

A recalled statement was judged functionally correct if and only if the recalled statement preserved the logic of the program and was a correct Fortran statement. Syntactic exactness to the original program statements was not considered necessary for correctness. For example, a program variable might have been referred to as *ND* and appeared in a statement as $ND = ND + 37$. If the subject recalled the statement as $L = L + 37$ and used L in every statement where *ND* originally appeared, those statements would be judged correct.

Results

Memory Organization and Algorithmic Logic Segments

If memory organization were based on logic segments, significantly more I/C or C/I transitions should occur at segment boundaries than nonsegment boundaries. The frequencies of C/I and I/C transitions at segment boundaries and at all nonboundaries were calculated. Seven Kolmolgorov-Smirnov (K-S) [22] tests were computed at the 0.01 level to test whether the frequencies of C/I and I/C transitions were significantly higher at boundaries than at nonboundaries. One test was conducted for each of the six experimental conditions and one test for all conditions combined.

All seven K-S tests resulted in significant differences between boundary and nonboundary statement transition patterns. In all seven tests there was a considerable higher combined percentage of C/I and I/C transitions at boundaries than nonboundaries. These percentages are presented in Table 1.

Memory Organization and Indentation

If indentation increases the correspondence between memory organization and a program's logic segments, the recalled boundary transition pattern of indented programs should contain more C/I and I/C transitions than the boundary pattern of unindented programs. The three unindented groups were combined into a single group. Likewise, the three indented groups were combined into a single group. A K-S test was conducted at the 0.01 level to determine whether the boundary transition patterns of indented programs contain significantly more I/C and C/I transitions than similar patterns of unindented programs. This test did not result in significant differences between the groups with indented and unindented programs.

The percentages of C/I and I/C transitions at segment boundaries for unindented and indented groups were 19 and 18%, respectively. The proportion of C/I and I/C transitions at boundaries indicates the degree to which memory organization corresponds to each program's logic segments. However, indentation did not increase the similarity between logical organization of the program and memory organization. The results of these tests provide evidence that indentation may not enhance memory organization.

Memory Organization and Documentation

If one or two documentation levels increases the relationship between memory organization and a program's logic segments, significantly more I/C or C/I transitions

TABLE 1. Combined percentages of C/I and I/C transitions.

	Indented		Unindented	
	Boundary	Nonboundary	Boundary	Nonboundary
No Documentation	19%	5%	21%	3%
Initial Documentation	22%	7%	15%	4%
Interspersed Documentation	17%	3%	18%	4%

should be observed in the boundary patterns of the respective groups. Three pairwise comparisons among the transition patterns for the three pairwise comparisons among the transition patterns, K-S tests did not result in significant differences among the three groups at the 0.01 level.

The percentages of C/I and I/C boundary transitions for groups with no documentation, initial documentation, and interspersed documentation were 20, 19, and 18%, respectively. These results indicate that documentation did not increase the extent to which memory organization reflected the logical organization of the program.

Indentation, Documentation, and Recall

A 2 × 3 Multivariate Analysis of Variance (MANOVA) was computed with indentation (present or absent) and documentation (none, initial, or interspersed) as the independent variables and each subject's percentages of correctly recalled lines in each of the five programs as the dependent variable. The analysis which is presented in Tables 2 and 3 indicated a significant multivariate documentation effect and a significant univariate documentation effect on Program 1. No other effects were significant. In order to specifically identify differing means, a Tukey *post hoc* test compared Program 1's means at the 0.01 significance level. This test indicated that the combined group with no documentation recalled significantly more correct lines than the combined groups with initial documentation and interspersed documentation.

Similar 2 × 3 MANOVA'S were computed for each program with the percentages of correctly recalled lines in each logic segment within a program as the dependent variables. This was done in order to examine in detail treatment effects at the beginning, middle, and end of programs. The results of these analyses paralleled the overall analysis.

Memory Organization and Recall

In order to examine the relationship between memory organization and recall, Pearson product moment correlations were computed between the number of correctly recalled lines and the number of I/C and C/I transitions at boundaries. The frequency of these transitions provided a measure of the degree to which memory organization resembled algorithmic organization of the program. The correlations which were computed for each group, for all programs, and for indentation and documentation levels are presented in Table 4. It can be noted in Table 4 that many of the correlations were not significantly different for 0. The correlation of −0.86 on Program 1 for Group 1 resulted from the fact that subjects in this group tended to recall most of the statements in Program 1. Consequently, in this instance, a high number of correctly recalled lines was associated with a low number of I/C and C/I transitions. However, sizable significant positive correlations were consistently observed for Programs 3 and 4. Since these programs were the most difficult, these analyses suggest that the relationship between memory organization and recall is more discernable as program complexity increases.

TABLE 2. Mean number of correctly recalled lines in each program by group.

Group	PRG1	PRG2	PRG3	PRG4	PRG5
UNIND, NO DOC	14.7	9.7	5.5	9.5	9.5
UNIND, INIT DOC	5.0	6.1	2.7	5.5	8.9
UNIND, INTER DOC	4.7	5.6	2.5	2.5	2.9
IND, NO DOC	6.3	5.4	3.7	7.5	7.3
IND, INIT DOC	5.5	6.6	2.1	8.5	8.0
IND, INTER DOC	8.4	9.5	5.2	6.8	7.5

TABLE 3. MANOVA and univariate *F* values.

Multivariate effects	*F*
Interaction	1.8
Documentation	2.2 $p < 0.019$
Indentation	0.900
Univariate Indentation Effects	*F*
Program 1	1.0
Program 2	0.01
Program 3	0.001
Program 4	0.79
Program 5	0.26
Univariate Documentation Effects	*F*
Program 1	4.53 $p < 0.015$
Program 2	0.14
Program 3	0.83
Program 4	1.34
Program 5	0.06

TABLE 4. Correlation between the number of correctly recalled lines and the frequency of C/I and I/C transitions at boundaries.

Group	PRG1	PRG2	PRG3	PRG4	PRG5
1	−0.86[a]	−0.04	0.71[a]	0.65[a]	−0.21
2	0.54	0.53	0.72[a]	0.19	0.31
3	0.82[a]	0.18	0.79[a]	0.92[a]	0.81[a]
4	0.30	0.09	0.81[a]	0.75[a]	0.20
5	0.70[a]	0.35	0.92[a]	0.73[a]	0.51
6	0.25	0.02	0.08	0.11	0.21
TOTAL	0.09	0.11	0.44[a]	0.51[a]	0.26[a]
Indented	0.29	0.02	0.23	0.41[a]	0.21
Unindented	−0.08	0.16	0.73[a]	0.55[a]	0.23
NO DOC	−0.24	0.21	0.72[a]	0.49[a]	0.02
INIT DOC	0.56[a]	0.41	0.71[a]	0.64[a]	0.40
INTER DOC	0.39	−0.17	0.24	0.13	0.36

[a]Significantly different from 0 at the 0.05 level.

Discussion

In sentence memory and recall experiments. I/C and C/I transitions have generally been considered as evidence of chunking or organization in memory. The literature indicates that the frequency of these transitions types in sentence recall experiments is significantly higher at phrase boundaries than nonboundaries. These findings suggest that natural language sentences are organized in memory according to the phrase structure of the grammar. The results of the present study indicate that algorithmic logic segments play a similar role in the memory organization of computer programs.

The data which are presented in Table 1 shows that the combined percentage of C/I and I/C transitions is three to seven times higher at algorithmic logic segment boundaries than nonboundaries, irrespective of experimental condition. When the statement transition patterns are analyzed for each experimental condition and for all subjects, the statement transition pattern at segment boundaries is always significantly different from the nonboundaries. C/I and I/C transitions are typically taken to indicate boundaries of an organized group of items in memory [13]. Consequently, the high percentage of C/I and I/C transitions at logic segment boundaries and the corresponding results of these tests provide evidence that algorithmic logic segments affect the organization of computer programs in human memory. Memory for computer programs, like memory for other meaningful materials as prose passages [4,7] circuit diagrams [5], and chess positions [2], appear to be organized in a way which reflects the logical structure of the material.

There are other reasons why the statement transition patterns could have occurred according to these distributions. It could be suggested that there was a large amount of guessing during recall. However even if this were true, the fact remains that there were more I/C and C/I "guesses" at boundary transitions than at nonboundary transitions. This could be interpreted that the subjects were keying on logic segment boundaries. Within a particular segment, they were either able to recall all the segment's statements or else they made many incorrect guesses. The important point is that the guessing pattern suggests organization according to algorithmic logic segments.

The findings indicate that indentation and documentation did not increase the similarity between memory organization and the logical organization of programs. The present findings are consistent with Love [15], Shneiderman and McKay [21], and Weissman [29,30]. These authors previously found nonsignificant differences in the number of correctly recalled lines between indented and unindented experimental conditions. The present study found no significant differences in segment boundary transition patterns between the respective groups which indicates that indentation may not aid memory organization of programs. If people typically organize information in memory according to a logical scheme as Miller [16] contends, it seems likely that computer programs are cognitively organized by some logic hierarchy. Logical indentation or documentation of program statements may not increase the degree of organization that takes place normally. The results of this experiment suggest that this is the case.

Surprisingly, the results seem to imply that indentation and documentation do not enhance the recall of computer programs. One possible factor could be that indentation might interfere with the visual image of the program. This could be especially true in this study due to the relatively short length of the programs. In longer programs indentation might aid the visual image through spatial cues. An examination of data reveals that all the groups had about the same percentage of C/I and I/C transitions at segment boundaries. That is, the extent to which memory organization paralleled program organization was constant for all groups.

The previous discussions concerning the lack of indentation and documentation effects on memory organization are appropriate concerning their effects on recall. If neither factor benefits memory organization, it reasonably follows that they should not necessarily increase recall. Also, since all subjects had the same amount of time to complete the task, subjects without documentation could have spent more time memorizing details of the program. Love [15], Shneiderman and McKay [21], and Weissman [29,30], also found that indentation did not significantly effect recall. The present study supports their findings and further suggests that the lack of an effect of indentation on recall may be related to the fact that indentation did not improve memory organization.

The fact remains that the present study found no difference between initial and interspersed documentation and previous studies did. There are at least three factors which may be responsible for this discrepancy. First, the program length could determine the value of any documentation. It may be that the longer the program is the more helpful the documentation becomes. Since programs in this study were all relatively short compared to those in other studies, the benefits of documentation could have been negated. Secondly, the experimental presentation time could have confounded the issue. Subjects who had documentation presumedly spent time reading it while studying the programs. Subjects without documentation spent the entire time examining the programs. This means that subjects with documentation had more to do in the same amount of time. Thirdly, and perhaps most importantly, documentation by its very nature is subjective. What is effective, meaningful documentation to one person might be less helpful to another. Because of this, more precise control of individual differences, programmer expertise, and documentation style appear desirable in order to adequately test the effects of computer program documentation.

In exploring the relationship between memory organization and recall, it is noteworthy that the correlations between the number of correctly recalled lines and the frequency of C/I and I/C transitions associated with Program 3 and 4 tended to be high, positive, and significant across each experimental condition. These two programs were the most difficult. In other words, memory organization was related to recall in the more complex programs. This finding is entirely consistent with previous memory organization studies [3,8,15,27,28] and may provide an indication of the conditions where a strong relationship between organization and recall can be expected [18].

Debugging of programs with logical errors in an activity which demonstrates the practical importance of memory organization processes underlying programmer behavior. If the programmer is questioning the correctness of the *N*th line of code, it is necessary that the preceeding statements, which are logically related to it, be readily available in the programmer's memory.

Suggested Further Research

This study presents evidence which suggests that memory organization is indeed important in the cognitive processing of computer programs. The relationship between organization and program logic might vary with respect to the ability of the programmer; in other words, the more experienced programmers might form chunks which comprise larger logic segments than novice programmers. The complexity of the program might also interact with the cognitive organization processes.

It would also be valuable to understand the relationship between memory organization and the time required to debug programs with logic errors. Presumedly, less time should be required to detect logical errors if the program's logic is well organized in memory.

Requests for reprints should be sent to Dr. A. F. Norcio, Applied Science Department, U.S. Naval Academy, Annapolis, MD 21402.

References

1. Ammon, P. R. "The perception of grammatical relations in sentences: A methodological exploration." *Journal of Verbal Learning and Verbal Behavior*. 7:869-875; 1968.
2. Chase, W. G.; Simon, H. A. "Perception in chess." *Cognitive Psychology*. 4:55-81; 1973.
3. Cofer, C. N. "On some factors in the organizational characteristics of free recall." *American Psychologist*. 20:261-272; 1965.
4. Crothers, E. J. "Memory structure and recall of discourse." In: R. O. Freedle and J. B. Carroll, Eds. *Language Comprehension and the Acquisition of Knowledge*. New York: Wiley; 1972.
5. Egan, D. E.; Schwartz, B. J. "Chunking in recall of symbolic drawings." Paper presented at the Annual Meeting of the American Educational Research Association, March, 1978.
6. Fodor, J; Beaver, T. "The psychological reality of linguistic segments." *Journal of Verbal Learning and Verbal Behavior*. 4:414-420; 1965.
7. Frederiksen, C. H. "Effects of task induced cognitive operations on comprehension and memory processes." In: R. O. Freedle and J. B. Carroll, Eds. *Language Comprehension and the Acquisition of Knowledge*. New York: Wiley; 1972.
8. Johnson, N. F. "The psychological reality of phrase structure rules." *Journal of Verbal Learning and Verbal Behavior*. 4:468-475; 1965.
9. Johnson, N. F. "The influence of associations between elements of structural verbal responses." *Journal of Verbal Learning and Verbal Behavior*. 5:368-374; 1966.
10. Johnson, N. F. "On the relationship between sentence structure and latency in generating the sentences." *Journal of Verbal Learning and Verbal Behavior*. 5:375-380; 1966.
11. Johnson, N. F. "The influence of grammatical units on learning." *Journal of Verbal Learning and Verbal Behavior*. 7:236-240; 1968.
12. Johnson, N. F. "The effect of a difficult word on the transitional error probabilities in sentences." *Journal of Verbal Learning and Verbal Behavior*. 8:518-523; 1969.
13. Johnson, N. F. "Chunking: Associative chaining versus coding." *Journal of Verbal Learning and Verbal Behavior*. 8:725-732; 1969.
14. Klatzky, R. L. *Human Memory: Structures and Processes*. San Francisco: W. H. Freeman; 1975.
15. Love, L. T. "Relating individual differences in computer programming performance to human information processing abilities." Ph.D. dissertation, University of Washington, 1977.
16. Miller, G. A. "The magical number seven, plus or minus two: Some limits on our capacity for processing information." *Psychological Review*. 63:81-97; 1956.
17. Miller, L. A. "Programming by non-programmers." *International Journal of Man-Machine Studies*. 6:237-260; 1974.
18. Shimmerlik, S. M. Organization theory and memory for prose: A review of the literature. *Review of Educational Research*. 48:103-120; 1978.
19. Shneiderman, B. *Measuring Computer Program Quality and Comprehension*. College Park: University of Maryland, Department of Information Systems Management, TR-16, 1977.
20. Shneiderman, B.; Mayer, R. *Towards a Cognitive Model of Programming Behavior*. Bloomington: Indiana University, Department of Computer Science, TR-37; 1975.
21. Shneiderman, B.; McKay, D. "Investigations of computer program debugging and modification." *Proceedings of the Sixth International Congress of the International Ergonomics Association*. 1976.
22. Siegel, S. *Nonparametric Statistics for the Behavioral Sciences*. New York: McGraw-Hill 1956.
23. Sime, M. E.; Green, T. R. G.; Guest, D. J. "Psychological evaluation of two conditional constructions used in computer languages." *International Journal of Man-Machine Studies*. 5:105-113; 1973.
24. Sime, M. E.; Green, T. R. G.; Guest, D. J. "Scope marking in computer conditionals—A psychological evaluation." *International Journal of Man-Machine Studies*. 9:107-118; 1977.
25. Sime, M. E.; Arblaster, A. T.; Green, T. R. G. "Reducing programming errors in nested conditionals by prescribing a writing procedure." *International Journal of Man-Machine Studies*. 9:119-126; 1977.
26. Suci, G. J. "The validity of the pause as an index of units in language." *Journal of Verbal Learning and Verbal Behavior*. 6:26-32; 1967.
27. Tulving, E. "Subjective organization in free recall of unrelated words." Psychological Review. 69:344-354; 1962.
28. Tulving, E. "Subjective organization and effects on repetition in multi-trial free recall learning." *Journal of Verbal Learning and Verbal Behavior*. 5:192-197; 1966.
29. Weissman, L. "A methodology for studying the psychological complexity of computer programs." Ph.D. dissertation, University of Toronto, 1974.
30. Weissman, L. "Psychological complexities of computer programs: An experimental methodology." *SIGPLAN Notices*. 9; 1974.