

Knowledge Organization and Skill Differences in Computer Programmers

KATHERINE B. MCKEITHEN AND JUDITH S. REITMAN
Bell Laboratories, Holmdel

AND

HENRY H. RUETER AND STEPHEN C. HIRTLE
The University of Michigan

Like experts in other fields, expert computer programmers can recall at a glance far more information relevant to their field than novices can. One explanation for this difference is that experts not only have more information, they have it better organized into meaningful chunks. In this paper, we infer the details of individual programmers' chunks of key programming concepts using the Reitman-Rueter (*Cognitive Psychology*, 1980, 12(4), 554-581.) technique for inferring tree structures from recall orders. Differences in organizations accompany skill-level differences. Beginner programmers' organizations show a rich variety of common-language associations to these programming concepts; Intermediate programmers show mixtures of programming and common-language associations; and Experts show remarkably similar, but not identical, organizations based clearly on programming knowledge.

Experts and novices differ in their abilities to process large amounts of meaningful information. This difference has been seen in the domains of chess, Go, electronics, bridge, music, and physics (Chase & Simon, 1973a, 1973b; Reitman, 1976; Sloboda, 1976; Egan & Schwartz, 1979; Engle & Bukstel, 1978; Charness, 1976; Larkin, McDermott, Simon, & Simon, 1980). A common explanation for this difference is that experts have not only *more* information, they have that information better organized into useful chunks. Instead of perceiving and remembering individual pieces of information, they process meaningful groups of information, making their perception more efficient and their recall performance much higher.

These earlier studies introduced the notion that superior organization of

This work is based in part on the Ph.D. dissertation of the first author, under the direction of Judith S. Reitman, and with the analytical assistance of Henry Rueter. The subsequent analyses of similarities among tree structures was contributed by Stephen Hirtle and Henry Rueter. This work was supported by NSF Grant BNS-76-82806 to J. S. Reitman at the University of Michigan. Requests for reprints should be sent to Judith S. Reitman, Bell Laboratories, 2B-617, Holmdel, NJ 07733.

information is the key to expert performance. They were weak, however, in specifying the details of that organization. For instance, both Chase and Simon, and Reitman assumed that chunks are separated by long pauses in recall. But identifying chunk boundaries with long pauses suffers from two drawbacks: First, pauses are intolerably variable across trials, and second, as used, reveal only a single level of chunking. This second shortcoming is particularly limiting since it is likely that the experts' chunks are hierarchical. Another way to uncover organization, used by Reitman, and Egan and Schwartz, is to present items on a sheet of paper and have the subject draw circles around those that are related. Though this drawing technique can delineate a hierarchical structure and even overlapping chunks, it is limited by the subject's ability to introspect accurately and reliably. Here, we use a new, more sophisticated technique for showing the details of the organizations of information for individual subjects and go on to describe the general pattern of differences in organization that accompany subjects' different skill levels.

The technique used here, developed by Reitman and Rueter (1980), produces a hierarchical representation of information from regularities in the orders in which items are recalled over many trials. This technique capitalizes on the fact that people tend to recall all items of one chunk before moving on to the next chunk (e.g., Cohen, 1966). Consequently, we define a chunk as a group of items recalled together on all trials. Similar techniques have been proposed, notably those of Friendly (1977), Monk (1976), and Bushke (Bushke, 1976; Zangen, Ziegelbaum, & Bushke, 1976), but they have quite different working assumptions. The Reitman and Rueter algorithm is preferred here because, unlike the others, it is based on a psychological theory of recall production and generates a structure that is unique, reliable, and detailed (see Reitman and Rueter (1980) for a detailed comparison with these other techniques).

Two benefits derive from using computer programmers as subjects. First, recent studies of the differences in expert and novice programmers, focusing on gross external performance measures, have met with mixed success. Skill or experience has been shown to be related to the time necessary to get a particular program written and running, the number of times a program is submitted before completion, the number of languages known, and the rated familiarity of certain programming concepts, but not with time to debug, modify, or comprehend a new program (Lucas & Kaplan, 1974; Sheppard, Curtis, Millman, & Love, 1979). Intuitively, one would expect that though some of these *external* measures may not differ, surely experts view these tasks differently than novices. This paper sets out to investigate *internal* differences.

The second benefit relates to a current debate in the computer science

literature on the merits of "structured programming." Structured programs are composed of a hierarchy of independent modules (see Shneiderman, 1980; Shneiderman & Mayer, Note 1). The advocates of structured programming hypothesize that such organizations of code are naturally easier to understand and modify because they are consonant with the programmer's mental organization. Though structured programs have been shown to be easier to modify than standard programs (Lucas & Kaplan, 1974), the argument that this is due to their consonance with mental organization is based only on expert intuition. The research here, though not a direct answer to that question, may provide an empirical base for testing these intuitions since it illustrates the use of a technique that provides a detailed representation of mental organization.

The work here has two parts. In our first study, the classic expert-novice difference in short-term recall is replicated with computer programmers who viewed either a coherent or a scrambled version of a computer program. This step is important in showing that the materials and experts that we studied give familiar results. Then in a second study, we infer the details of subjects' organizations of programming concepts by constructing hierarchical representations of the relations among computer language keywords. The similarities and dissimilarities of these inferred organizations are then related to the subjects' known skill-level differences.

EXPERIMENT I

Method

Subjects of three skill levels were shown a 31-line, ALGOL W computer program in either normal or scrambled version for five 2-min study trials. Groups of two or three subjects saw the program projected onto a screen by an overhead projector. In the 3-min recall period that followed each trial, subjects wrote as much of the program as they could remember on a blank recall sheet, putting each recalled line on the sheet as close to its presented position as possible. A blank recall sheet was used for each trial; subjects were prevented from looking at their previous recall attempts and were asked to recall everything anew on each trial.

The program to be recalled was written by an ALGOL W expert, the third author, who did not serve as a subject in this experiment. Tables 1 and 2 contain the actual program; the first is the normal, executable version, the second a randomly scrambled version. Since indentation is important to the meaning of an ALGOL W program, it was retained in the scrambled version. Half of the subjects in each skill level saw the normal version, half the scrambled.

Fifty-three subjects at three skill levels were tested in Experiment I. Twenty-four subjects, called Beginners, were just starting their first ALGOL W course; six of these had previous experience with either BASIC or FORTRAN. Twenty-three subjects, called Intermediates, had just completed their first ALGOL W course; seven of these had previous experience with either BASIC or FORTRAN. Six subjects, the Experts, taught ALGOL W, had over 2000 hr of general programming experience, and had an average of over 400 hr of experience in ALGOL W.

Results

The recall sheets were scored by counting the number of lines written verbatim in their proper relative order. To be counted as correct, a line needed neither to be placed in its exact position on the recall sheet nor to have the proper indentation. A simple order reversal of lines was scored as just one correct. The only shorthand allowed was a ditto mark.

Figure 1 shows the number of lines correctly recalled on each of the five trials by Beginners, Intermediates, and Experts after seeing either the normal or scrambled version. Clearly, skill level differentiates recall performance for the normal version but not for the scrambled version. Unlike the results of Chase and Simon, and Reitman for chess and Go, the skill-related differences appear on more than just the first trial.

The results of a three-way analysis of variance confirm this pattern of results. Significance is accepted at $p < .01$. There were significant main effects for trials, $F(4, 188) = 108.01$; version (normal or scrambled), $F(1, 188) = 25.63$; and skill level, $F(2, 188) = 11.84$. All interactions were likewise significant: version \times trial, $F(4, 188) = 3.32$; trial \times level, $F(8, 188) = 4.99$; version \times level, $F(2, 188) = 15.7$; and version \times trial \times level, $F(8, 188) = 2.71$.

The two-way interaction involving skill level and version suggests that experienced programmers show superior recall only in the normal version

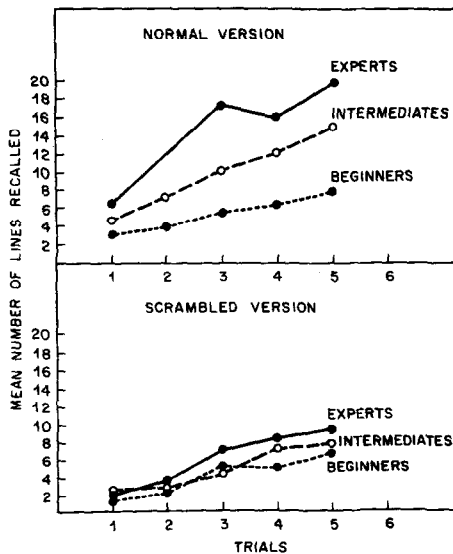


FIG. 1. Number of program lines recalled by Experts, Intermediates, and Beginners after being presented with either a normal version or a scrambled version of the program.

of the program. A *t* test between Experts' and Beginners' recall, averaged over trials, was not significant for the scrambled version but was significant for the normal version, $t(14) = 2.4, p < .05$. Further, the three-way interaction suggests that for the normal version the disparity in recall among subjects of differing experience changed over trials. Figure 1 shows that the disparity increased with added trials. Only this last result differs from the results of Chase and Simon and Reitman, where most of the skill difference appeared on the first trial; there, the disparity *decreased* with added trials.

Closer examination of the particular lines recalled reveals characteristic patterns of recall for subjects with different skill levels. The right-hand panels of Tables 1 and 2 show the proportions of subjects at each skill level who recalled each line on the last (fifth) trial. In the normal version of the program (Table 1), there are clear differences. Although subjects of all skill levels recalled the short lines, such as the initial BEGIN and the

TABLE 1
Proportion of Experts (E), Intermediates (I), and Beginners (B)
Who Recalled Each Line of the Normal Version of the Program

	E	I	B
1 BEGIN	1.00	.83	.83
2 INTEGER PTR, TRIALS, OBJECTS;	1.00	.83	.83
3 INTEGER ARRAY ORDERS (1::32, 1::21);	1.00	.67	.50
4 INTEGER ARRAY TEMP (1::32);	.67	.67	.50
5 STRING (27) ALPHA;	.67	.75	.50
6 ALPHA := "ABCDEFGHIJKLMNQPQRSTUVWXYZ";	.67	.83	.58
7 READ (TRIALS, OBJECTS);	.67	.75	.33
8 ASSERT (0 < TRIALS AND TRIALS ≤ 32);	.33	.58	.50
9 ASSERT (0 < OBJECTS AND OBJECTS < 27);	.33	.67	.33
10 FOR I := 1 UNTIL TRIALS + 1 DO	1.00	.83	.00
11 FOR J := 1 UNTIL OBJECTS DO	1.00	.92	.17
12 BEGIN READON (ORDERS (I,J));	.67	.67	.17
13 ASSERT (0 < ORDERS (I,J) AND ORDERS (I,J) < 27)	.33	.25	.00
14 END;	.67	.58	.50
15 FOR J := 1 UNTIL OBJECTS DO	.67	.67	.08
16 IF ORDERS (TRIALS + 1,J) ≥ J THEN	.33	.33	.00
17 BEGIN PTR := 1;	.33	.17	.08
18 WHILE ORDERS (TRIALS + 1,PTR) ≥ J DO PTR := PTR + 1;	.33	.00	.00
19 ASSERT PTR < 27;	.33	.08	.00
20 FOR I := 1 UNTIL TRIALS + 1 DO	.67	.17	.00
21 BEGIN TEMP(I) := ORDERS (I,J);	.33	.08	.08
22 ORDERS (I,J) := ORDERS (I,PTR);	.67	.08	.00
23 ORDERS (I,PTR) := TEMP (I)	.33	.08	.00
24 END	.67	.25	.08
25 END;	.67	.33	.17
26 FOR I := 1 UNTIL TRIALS DO	.67	.25	.00
27 BEGIN WRITE (I, " ");	.67	.08	.00
28 FOR J := 1 UNTIL OBJECTS DO	.67	.17	.08
29 WRITEON (ALPHA (ORDERS (I,J) 1))	.33	.00	.00
30 END	.33	.00	.00
31 END.	.33	.67	.67

TABLE 2
 Proportion of Experts (E), Intermediates (I), and Beginners (B)
 Who Recalled Each Line of the Scrambled Version of the Program

	E	I	B
1 WHILE ORDERS (TRIALS + 1,PTR) ≥ J DO PTR := PTR + 1;	.33	.36	.33
2 READ (TRIALS, OBJECTS);	.67	.36	.42
3 BEGIN PTR := 1;	1.00	.64	.50
4 ASSERT (0 < ORDERS(I,J) AND ORDERS(I,J) < 27)	.33	.45	.17
5 FOR I := 1 UNTIL TRIALS DO	1.00	.64	.33
6 END	.67	.45	.67
7 ASSERT (0 < OBJECTS AND OBJECTS < 27);	1.00	.36	.00
8 IF ORDERS (TRIALS + 1,J) ≥ J THEN	.67	.00	.08
9 ASSERT (0 < TRIALS AND TRIALS ≤ 32);	.33	.09	.25
10 BEGIN WRITE (I, " ");	.00	.18	.25
11 INTEGER ARRAY ORDERS (1::32, 1::27);	.00	.27	.00
12 END;	.00	.09	.33
13 FOR I := 1 UNTIL TRIALS + 1 DO	.33	.09	.17
14 ORDERS (I, PTR) := TEMP (I)	.00	.09	.00
15 STRING (27) ALPHA;	.67	.55	.08
16 BEGIN READON (ORDERS (I,J));	.00	.00	.00
17 ALPHA := "ABCDEFGHJKLMNOPQRSTUVWXYZ";	.67	.55	.50
18 END.	.67	.45	.42
19 FOR J := 1 UNTIL OBJECTS DO	.00	.27	.17
20 BEGIN TEMP(I) := ORDERS (I,J);	.00	.00	.00
21 FOR J := 1 UNTIL TRIALS + 1 DO	.00	.36	.00
22 END;	.00	.09	.17
23 BEGIN	.33	.18	.50
24 FOR J := 1 UNTIL OBJECTS DO	.00	.18	.17
25 ASSERT PTR < 27;	.00	.00	.00
26 INTEGER PTR, TRIALS, OBJECTS;	.00	.00	.00
27 WRITEON (ALPHA (ORDERS (I,J) 1))	.00	.00	.00
28 ORDERS (I,J) := ORDERS (I,PTR);	.00	.00	.08
29 INTEGER ARRAY TEMP (1::32);	.00	.18	.17
30 FOR J := 1 UNTIL OBJECTS DO	.33	.18	.33
31 END	.33	.45	.75

ENDs scattered throughout the program (lines 1, 14, 24, 25, and 31, respectively), the Experts and Intermediates added the familiar lines for the beginning of a nested loop for reading in a matrix (lines 10, 11, and 12), and the Experts added parts of a later loop that exchanges values and begins the output sequence (lines 20–28). The Experts have used similar devices many times, recognize the lines as a chunk, and are therefore able to recall more.

In the scrambled version of the program (Table 2), skill differences are not so apparent. Subjects of all skill levels tended to recall the lines that set up the string of alphabetic characters (lines 15 and 17), a few of the middle ENDs (lines 6 and 18 but not 12 and 22), and the END that fortuitously appeared at the end (line 31). But only the Experts additionally recalled the familiar loop constituents near the beginning (lines 3, 5, and 7), missing the easier final END (line 31). In the scrambled version, the correlations among particular lines recalled on the fifth trial were large

for all skill levels ($E-I = .72$, $E-B = .47$, $I-B = .57$, all $p < .01$); but in the normal version, correlations were large only between the adjacent skill groups, and not between Experts and Beginners ($E-I = .57$, $I-B = .69$, $p < .01$, and $E-B = .23$, $p < .10$).

We have replicated the correlation between expertise and recall that has been found in similar studies of expert–novice differences, both in computer programming (Love, 1977) and in other domains. The next step in understanding the basis for this superior recall is to examine the differences in the way subjects organize their recall.

EXPERIMENT II

A rich source of low-level concepts that are useful in understanding the structure of a program is found in the language's reserved words. Reserved words specify particular, important functions in the language and thus cannot be used for other purposes by the programmer. Consequently, ALGOL W reserved words were selected for use in the second experiment. Using the Reitman and Rueter technique, we inferred individual subjects' organizations of the reserved words, expecting to find differences that are related to skill level.

Method

Subjects of the same three skill levels as in Experiment I were run individually in a free-learning, free-recall task. They were each given a shuffled deck of twenty-one 3×5 in. cards on which were written the ALGOL W reserved words selected for this experiment, one word per card. Subjects were asked to learn the words so that they could recall them aloud without the aid of the cards. It was suggested that sorting the cards might help them learn the words more easily, and all subjects did so while learning. After the subjects went through their own self-paced, self-directed study–test procedures, they were given some practice trials. On most of these trials, subjects recalled as they wished ("noncued" trials); on a few others ("cued" trials), they were asked to start with a given word, continue with those "that go with it," and recall the remainder. The data collection phase began *after* the subjects had correctly recalled all 21 words on two consecutive practice trials.

In this phase, the subjects were tape-recorded as they recalled all 21 words 25 times. On 21 trials they were "cued" with one of the ALGOL W words, each word serving as a cue on one trial; on the rest of the trials, they were not cued. If recall on a particular trial was in error, that trial was repeated at the end of the scheduled 25 trials. Because recall times were being collected to validate the Reitman and Rueter technique, only those ALGOL W reserved words with one syllable were used. (See McKeithen (1979) for a discussion.) Additionally, four abbreviations of polysyllabic words, such as ABS and DIV, were excluded because of the pronunciation difficulties that they might have caused. The final set of words is listed at the "leaves" of the trees in Figs. 3–8.

Twenty-two subjects served in this experiment: eight Experts, six Intermediates, and eight Beginners. The Experts and three of the Beginners had participated in Experiment I. The new Beginners and Intermediates fit the skill levels as described in Experiment I. Four of the six new Intermediates had previous experience with either BASIC or FORTRAN.

Results

The recall orders from each subject were analyzed using the Reitman and Rueter algorithm (1980). This algorithm searches all of a subject's recall strings for all groups of items that always appeared contiguously, regardless of order. For example, in the four strings at the top of Fig. 2, the nontrivial groups of contiguously recalled items are ABC, BC, DEFG, DEF, EFG, DE, EF, and FG. These groups form a lattice under set inclusion (Fig. 2b); the lattice can be represented either as an ordered tree (Fig. 2c) or as a parenthesized expression (Fig. 2d). Consistency in the order of recall of groups (such as a list or an ordered set of chunks) appears in the lattice as overlapping groups; e.g., DEFG, DEF, EFG, DE, EF, and FG indicate that DEFG was always ordered. Determination of whether the order was always the same (a "unidirectional chunk") or one order and its reverse (a "bidirectional chunk") requires inspection of the original recall strings. In our example above, DEFG is a unidirectional chunk. Unidirectional chunks are indicated on the trees by single-headed arrows over the constituent branches and in the parenthetic expression by square brackets. Bidirectional chunks are indicated by double-headed arrows in the trees and angle brackets in the parenthetic representation. "Nondirectional chunks," those whose constituents appear in any order,

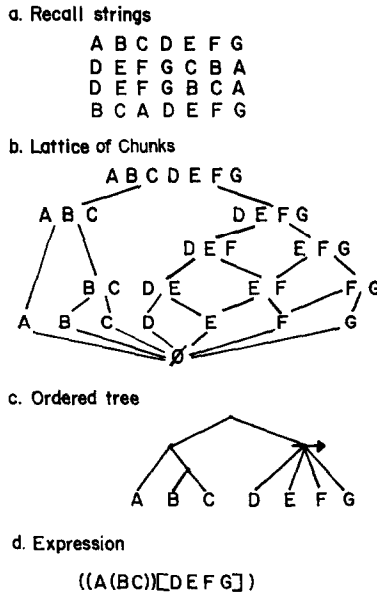


FIG. 2. Example analysis of recall strings by the Reitman-Rueter technique.

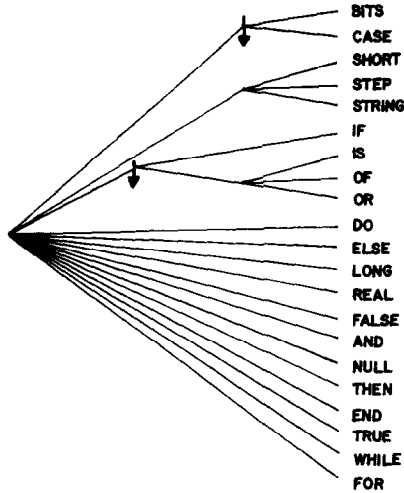


FIG. 3. A Beginner's organization of 21 ALGOL W reserved words, apparently based on orthography (Subject 14).

are indicated by the absence of an arrow in the tree and by curved parentheses in the parenthetic expression.

In the following, the organizations obtained for the 22 subjects are reported first by example and then by more fine-grained analyses of all subjects.

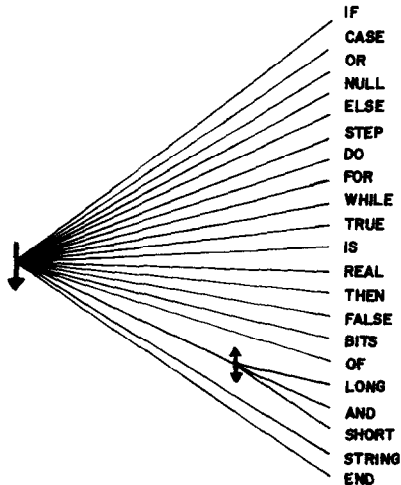


FIG. 4. Another Beginner's organization of 21 ALGOL W words, apparently based on a story mnemonic (Subject 13).

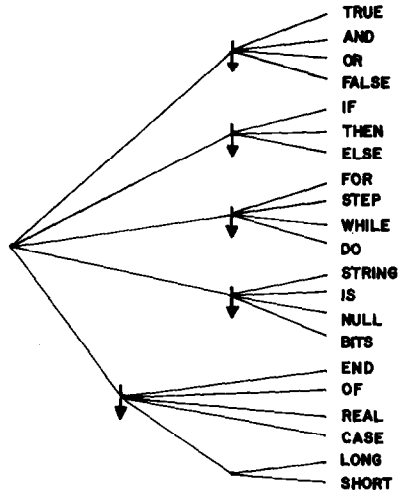


FIG. 5. An Intermediate's organization of the ALGOL W words, showing a mixture of ALGOL and common-language chunks (Subject 2).

Figures 3–8 show six trees typical¹ of the kinds of structures obtained for subjects at each skill level, two for Beginners (Figs. 3 and 4), two for Intermediates (Figs. 5 and 6), and two for Experts (Figs. 7 and 8). The Beginners appear to have used a variety of general mnemonic techniques. In Fig. 3, the chunks in which the words begin with the same first letter or have the same length indicate that this Beginner was using a mnemonic technique based on orthography. In the large unidirectional tree in Fig. 4, that of another Beginner, the common language sequences such as DO-FOR-WHILE, TRUE-IS-REAL-THEN-FALSE, and BITS-OF-(LONG-AND-SHORT)-STRING indicate story construction, another frequent mnemonic technique for Beginners.

The Intermediates, having seen these words in the context of the ALGOL W course they just completed, showed more clustering specific to meaning in ALGOL W. In Fig. 5, the IF-THEN-ELSE and FOR-STEP-WHILE-DO chunks are evidence for programming knowledge, whereas the STRING-IS-NULL-BITS and END-OF-REAL-CASE appear to be natural language sequences. In Fig. 6, another Intermediate, IF-THEN-ELSE and AND-OR-TRUE-FALSE-BITS are ALGOL W

¹ Subjects selected as examples were those whose organizations were easy to characterize and appeared to be representative of their skill group. Afterward, we noticed that they are prototypical in another sense: They all fell in the middle of their skill-level cluster on the vertical "Expert" dimension in Fig. 9 (described later in the paper). These six subjects are marked in this figure with "feet," ovals drawn where their stems hit the reference grid.

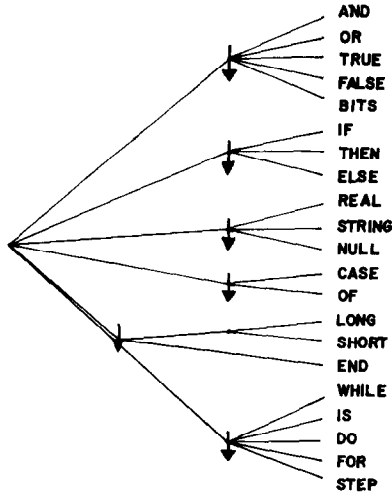


FIG. 6. Another Intermediate's organization, again showing a mixture of ALGOL and common-language chunks (Subject 21).

chunks, but the others are less clear. It appears as if the Intermediates' organizations show a *mixture* of ALGOL W and natural-language chunks.

The Experts' structures show a great deal of clustering according to meaning in ALGOL W. In Figs. 7 and 8, WHILE-DO and FOR-STEP cluster for their function in loops. In Fig. 7, STRING-BITS-LONG-

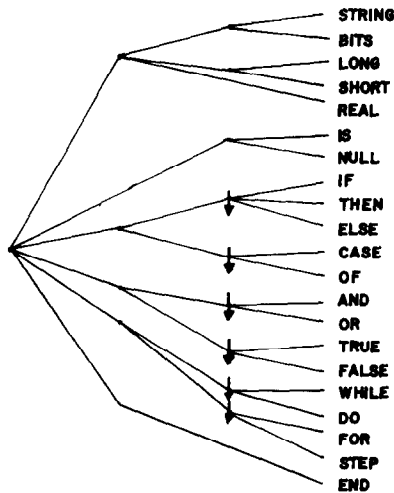


FIG. 7. An Experts' organization, showing chunking based on ALGOL meaning (Subject 1).

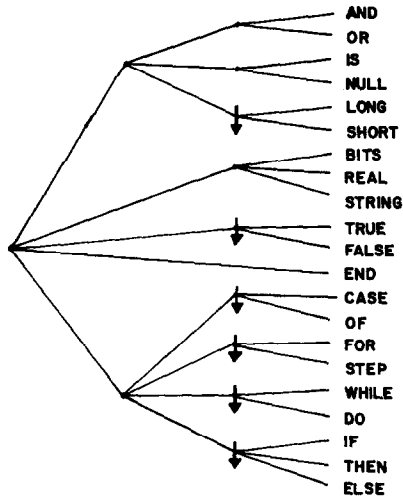


FIG. 8. Another Experts' organization, again showing ALGOL-based chunking (Subject 4).

SHORT-REAL cluster for their descriptive function of data types. And, in Fig. 8, the four chunks, CASE-OF, FOR-STEP, WHILE-DO, and IF-THEN-ELSE, all cluster because they are the words that control the flow of the program, the others being more static or descriptive in nature. Only END is unchunked in these trees; its logical mate, BEGIN, was eliminated from the recall set because it has more than one syllable.

The six trees just discussed in detail were picked to be representative of their skill-level groups. All subjects figure in the analyses that follow. We examine three questions: Are Experts *more* organized than Beginners? Do the depths of the organizations differ across the skill groups? Do organizations within a skill group look alike?

Are Experts more organized? Bushke (1976) and others have shown that the more chunking one does, the better the recall. It may be, then, that our Experts in Experiment I recall better because they have more organization, more chunks, or more ordering between items or chunks. To answer this, we need to attach to each subject's ordered tree a number reflecting the *amount* of organization. Though the literature offers several measures of organization based on the number of items appearing contiguously on successive trials (e.g., Tulving, 1962; Pellegrino, 1971), we adopted one that is also sensitive to the multilevel and order information in these ordered trees.

The measure is called PRO for "possible recall orders." It is a function of the number of different recall orders that could be generated from the

inferred tree. In general, the smaller the PRO, the more organization in the structure. (See Reitman & Rueter (1980) for a discussion.)

The Experts had a mean PRO of 6.8, Intermediates 14.6, and Beginners 7.8. None of the differences was significant, primarily because of the high variability among subjects (i.e., the range of PROs was 45). At least for this simple measure of the amount of organization, there was no difference among skill levels.

Do the depths of the organizations increase with skill level? If a subject normally organizes information in chunks of chunks, he is likely to be able to recall better in a short-term recall task because the code held in STM, a higher-level node, refers to more information, that from several chunks. Chase and Simon, finding that their Experts revealed more chunks than beginners though they showed no larger a STM capacity, argued similarly that Experts' chunks might be hierarchically organized. It may be that our Experts' organizations are deeper, more nested, than Beginners'. The Expert in Fig. 8, for example, has a deeper organization than does the Beginner in Fig. 4. If we examine all 22 subjects, however, we find that the average number of nodes between root and terminal items for Experts is 2.26, for Intermediates it is 1.79, and for Beginners it is 1.88. None of these differences is significant. Depth of organization does not seem to increase with skill level.

Do subjects within a skill group organize the particular words in similar ways? The above results suggest that it is unlikely that the overall pattern of organization provides the advantage in recall performance. More likely, it is the *content* of these organizations, the match between the way these concepts are stored and the way they appear in the program to be recalled. We would expect Experts' organizations to be similar, based on useful programming relationships. And, we might expect Intermediates' and Beginners' organizations to be similarly clustered within their skill group but based on relationships less useful in recalling a program.

What did our subjects' organizations have in common? Table 3 lists all of the chunks common to four or more subjects of any skill level. In the listing, the chunks have been grouped according to the distribution of occurrences in the skill levels. LONG-SHORT, TRUE-FALSE, and the other chunks listed at the top of the table are nearly equally represented in all skill levels, presumably because of their common associations in ordinary discourse and ALGOL W. IF-THEN-ELSE, CASE-OF, FOR-STEP, WHILE-DO, and those listed in the middle are chunks common only to Experts and some of the Intermediates, presumably because they are ALGOL W-based chunks. The chunk listed at the bottom of the table was the *only* chunk common to Beginners but not Intermediates or Experts. This list indicates that Experts and some Intermediates shared

TABLE 3
 Proportion of Experts (E), Intermediates (I), and Beginners (B)
 Whose Trees Include These Chunks

	E	I	B
Chunks evenly represented by all levels			
TRUE FALSE	1.00	.50	.62
LONG SHORT	.75	.67	.62
STRING BITS	.62	.17	.38
TRUE FALSE AND OR	.25	.33	.12
DO FOR STEP	.25	.50	.12
WHILE DO FOR STEP	.25	.50	.12
DO STEP	.12	.17	.25
Chunks more common for Experts			
IF THEN ELSE	.88	.50	.12
IF THEN	.88	.50	.38
THEN ELSE	1.00	.50	.12
AND OR	.88	.83	.25
CASE OF	1.00	.50	.00
FOR STEP	.88	.67	.00
REAL STRING BITS	.75	.17	.00
WHILE DO	.62	.33	.12
REAL STRING	.50	.17	.00
IS NULL	.38	.17	.12
LONG SHORT REAL STRING BITS	.50	.00	.00
LONG SHORT REAL	.50	.00	.00
Chunk more common for Beginners			
All but END ^a	.00	.00	.50

^a END was recalled either first or last.

some aspects of their organizations, presumably ALGOL associations, and that Beginners shared very few aspects, presumably choosing different common-language associations from the many available.

The homogeneity of organizations within skill levels can be assessed formally. What is necessary is a measure of the similarity of each pair of trees, and a more global analysis of all these pairwise similarities that reveals clusters of similar organizations. These clusters would then be examined for their correspondence to subjects' known skill levels.

The required global analysis of similarities is readily available from multidimensional scaling (Shepard, 1962a, 1962b). Measures of the similarity between two *ordered* trees, however, were not so readily available. The "cophenetic correlation" between pairs of trees' item-distance matrices (Sneath & Sokal, 1973) appeared, but it was dismissed because it failed to capture the order information in our trees (a fully nondirectional

bush and a fully ordered list would be identified as maximally similar). Consequently, we devised our own measure, one that is appropriate to Reitman–Rueter-ordered trees as well as ordinary hierarchies (such as those produced by Johnson's (1967) hierarchical clustering scheme).

Our measure of the similarity of two trees is based on the number of nontrivial chunks the two trees have in common—chunks at all levels in the tree, excluding single items and the group of all items. Thus, the tree $((AB)(CD))EFG$ shares one chunk with $((ABCD)(EFG))$, i.e., $ABCD$, and three with $((AB)(CD))(E(FG))$, i.e., AB , CD , and $ABCD$.²

More precisely, we define the similarity between two trees as:

$$\frac{\ln(\text{the number of chunks the two trees have in common} + 1)}{\ln(\text{the total number of chunks contained in both trees} + 1)}$$

By dividing the number of common chunks by the total, we construct a proportion in common. The greater the similarity, the greater the proportion of chunks two trees share. The logarithmic transformation eliminates the significant correlation found between the total number of chunks and the ratio. Distance is then given by $1 - \text{similarity}$.

To compare the Experts', Intermediates', and Beginners' organizations, we first filled a distance matrix with the distances between each pair of trees as described above, then applied multidimensional scaling. The best solution (stress = .093) was the three-dimensional configuration shown in Fig. 9, where the trees of the individual subjects are labeled by their skill level: Experts as circles, Intermediates as squares, and Beginners as triangles. In this picture, subjects who shared many chunks (and thus organized the words similarly) appear close together. Pairs of subjects who had dissimilar organizations, e.g., an alphabetizer and a storyteller, appear far apart. If Experts had organized in ways similar to each other, and Beginners and Intermediates in ways like their cohorts but unlike Experts, this figure would have shown three separate clusters.

The actual configuration does not have three separate clusters ordered on an "Expert" dimension; it is conical, with the Experts close to each other at the apex, and most of the less skilled subjects placed lower on the

² In counting the number of chunks in common when a directional chunk is involved, all overlapping subchunks implicit in the directional chunk (and elements of the lattice of all chunks) are included. Thus, $[ABCD]$ has two of its five nontrivial chunks in common with $((AB)(CD))$, i.e., AB and CD , but not BC , ABC , or BCD , and one chunk in common with $((ABC)D)$, i.e., ABC , but not BCD , AB , BC , or CD . There is *no* distinction in this measure between uni- and bidirectional chunks.

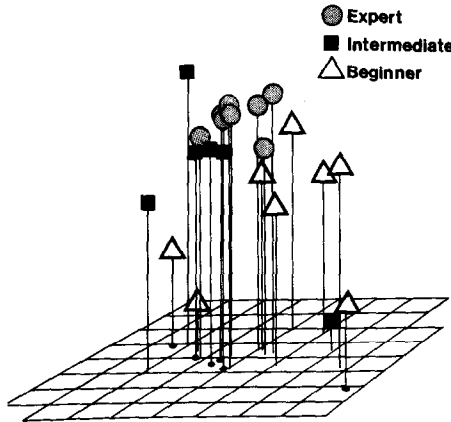


FIG. 9. The multidimensional scaling configuration of the distances between subjects' organizations.

vertical "Expert" dimension and spread further apart.³ It seems meaningful to label the vertical dimension but not the other two. They were simply necessary to adequately separate the disparate Beginners and Intermediates.

Table 4 shows the average distances within and across skill-level groups. Experts are more cohesive as a group than either of the other two groups. Furthermore, the Intermediates and Beginners are on average slightly closer to the Experts than they are to each other.

DISCUSSION

The classic expert-novice difference in short-term recall of meaningful material was replicated in the domain of computer programming. Unlike the similar studies in other domains, our Experts' superiority was not restricted to the first trial; they continued to recognize familiar segments of code after the first trial, so their performance diverges from that of the lesser skilled subjects. This difference may arise from the fact that the pictorial representations in chess, Go, bridge, and electronics diagrams allow Experts to recognize familiar chunks at a glance, whereas the computer experts required several trials to understand the program's various functions, recognize its overall structure, and then recall more chunks—on each successive trial.

Having demonstrated this recall difference, we went on to delineate the

³ The two striking outliers in this configuration are Intermediate subjects, one in the upper left and one at the lower right. The upper subject has very little organization; only three pairs of items were chunked. The subject at the lower right was the only subject to have completely alphabetized the set of words.

TABLE 4
Average Distance between Subjects within and across Skill Groups

	Experts	Intermediates	Beginners
Experts	.52	.61	.79
Intermediates		.73	.80
Beginners			.81

particulars of the organizations of computer concepts that likely contribute to this task. All subjects memorized the computer language keywords, but the strategies they brought to bear on the memorization task differed. Finally—and most important—these different strategies were related to skill level.

Beginners, ignorant of the functional significance of these words, appeared to adopt very general mnemonic techniques for memorizing and recalling the words. Typical were chunks of words with common first letters and lengths, words alphabetized by first letters, and small common-language sequences that are probably parts of stories. The number of possible organizations of these words based on common associations is large; each word by itself has a variety of associations, and different ones are selected depending on contexts provided by the subject's current thoughts or by the words next to it in that trial's presentation. Beginners shared at most only a few chunks with each other, resulting in large distances between them in the multidimensional scaling configuration.

The Intermediates were more similar to each other in their organizations, presumably because they all had recent common experience with the ALGOL W meaning of these words. They probably used these associations during the experiment and thus shared more chunks, both with each other and with the Experts. In many of these subjects' organizations, however, we see some common-language sequences, indicating that not all the recently learned computer associations were as strong as prior ones. Thus, REAL-CASE occasionally loomed stronger than CASE-OF, and BITS-OF-STRING stronger than nonnumeric data types STRING and BITS.

The Experts' associations from long-term experience with these words in programming predominated over common-language associations. Their organizations contained words grouped according to their function in ALGOL W, and resulted in a set of remarkably similar organizations that were clustered tightly in the multidimensional scaling configuration. That Intermediates were more similar to the Experts than to each other attests

to the convergence of their associations to those capturing the functional meaning in ALGOL W.

It is easy to see how this functional organization helps the Experts recall the computer program in the short-term recall task of Experiment I. If the subject knows that BEGIN goes with END, that WHILE(FOR-STEP)DO initiates loops, etc., then whenever he encounters one constituent of a chunk he expects the other. Subjects then either look for these in the program, or assume they are there and guess them in recall. These associations are the pegs on which the details of the program hang, like the classic mnemonic pegs of the Method of Loci and the Pegword System (Bower, 1970).

What we have shown here, despite complicated preliminaries, is simply a correlation between expertise and particular mental organizations of concepts in the field of expertise. It is tempting to conclude that this organization *produces* expertise. Though this may be true, these data do not prove it; they merely exhibit that subjects with an *existing* skill level have a particular common organization. To conclude that learning this organization would *increase* skill requires a demonstration of within-subject skill changes. However, our intuitions that acquisition of well-ordered knowledge produces greater skill, reflected in our educational principles, would make such a demonstration unsurprising.

The work here does suggest that the application of the Reitman-Rueter technique to Experts' recall of large programs might be useful to the study of "structured programming." A structured program is a hierarchy of independent possibly ordered chunks. If Experts' mental organizations match that hierarchy, arguments about cognitive advantages of that style of programming would be empirically rather than intuitively based.

Finally, the procedure introduced here for measuring the similarity of two tree structures based on the proportion of chunks they have in common generated results that were meaningful and interesting. There were clear, easily interpreted patterns of skill difference in our multi-dimensional scaling configuration. Though the specific limits of the measure have not yet been fully explored, it has face validity and is applicable to both Reitman-Rueter and other hierarchical representations.

REFERENCES

- Bower, G. H. Analysis of a mnemonic device. *American Scientist*, 1970, 58, 496-510.
- Bushke, H. Learning is organized by chunking. *Journal of Verbal Learning and Verbal Behavior*, 1976, 15, 313-324.
- Charness, N. Memory for chess positions: Resistance to interference. *Journal of Experimental Psychology: Human Learning and Memory*, 1976, 2, 641-653.
- Chase, W. G., & Simon, H. A. Perception in chess. *Cognitive Psychology*, 1973, 4, 55-81. (a)

- Chase, W. G., & Simon, H. A. The mind's eye in chess. In W. G. Chase (Ed.), *Visual information processing*. New York: Academic Press, 1973. (b)
- Cohen, B. H. Some-or-none characteristics of coding behavior. *Journal of Verbal Learning and Verbal Behavior*, 1966, 5, 182-187.
- Egan, D. E., & Schwartz, B. J. Chunking in recall of symbolic drawings. *Memory and Cognition*, 1979, 7, 149-158.
- Engle, R. W., & Bukstel, L. Memory processes among bridge players of differing expertise. *American Journal of Psychology*, 1978, 91, 673-689.
- Friendly, M. L. In search of the M-gram: The structure of organization in free recall. *Cognitive Psychology*, 1977, 9, 188-249.
- Johnson, S. C. Hierarchical clustering schemes. *Psychometrika*, 1967, 32, 241-254.
- Larkin, J., McDermott, J., Simon, D. P., & Simon, H. A. Expert and novice performance in solving physics problems. *Science*, 1980, 208, 1335-1342.
- Love, L. T. *Relating individual differences in computer programming performance to human information processing abilities*. Unpublished doctoral dissertation, University of Washington, 1977.
- Lucas, H. C., Jr., & Kaplan, R. B. A structured programming experiment. *Computer Journal*, 1974, 19, 136-138.
- McKeithen, K. B. *Assessing knowledge structures in novice and expert programmers*. Unpublished doctoral dissertation, University of Michigan, 1979.
- Monk, A. F. A new approach to the characterization of sequential structure in multitrial free recall using hierarchical grouping analysis. *British Journal of Mathematical and Statistical Psychology*, 1976, 29, 1-18.
- Pellegrino, J. W. A general measure of organization in free recall for variable unit size and internal sequential consistency. *Behavioral Methods and Instrumentation*, 1971, 3, (5), 241-246.
- Reitman, J. S. Skilled perception in Go: Deducing memory structures from interresponse times. *Cognitive Psychology*, 1976, 8, 336-356.
- Reitman, J. S., & Rueter, H. R. Organization revealed by recall orders and confirmed by pauses. *Cognitive Psychology*, 1980, 12(4), 554-581.
- Shepard, R. N. The analysis of proximities: Multidimensional scaling with an unknown distance function. I. *Psychometrika*, 1962, 27, 125-140. (a)
- Shepard, R. N. The analysis of proximities: Multidimensional scaling with an unknown distance function. II. *Psychometrika*, 1962, 27, 219-246. (b)
- Sheppard, S. B., Curtis, B., Millman, P., & Love, T. Modern coding practices and programmer performance. *Computer*, 1979, 12(12), 41-49.
- Shneiderman, B. *Software psychology: Human factors in computer and information systems*. Cambridge, MA: Winthrop, 1980.
- Sloboda, J. A. Visual perception of musical notation: Registering pitch symbols in memory. *Quarterly Journal of Psychology*, 1976, 28, 1-16.
- Sneath, P. H. A., & Sokal, R. R. *Numerical taxonomy*. San Francisco: Freeman, 1973.
- Tulving, E. Subjective organization in free recall of "unrelated" words. *Psychological Review*, 1962, 69, 344-354.
- Zangen, M., Ziegelbaum, P., & Bushke, H. CLUSTER: A program for identifying recurrent clusters. *Behavior Research Methods and Instrumentation*, 1976, 8(4), 388.

REFERENCE NOTE

1. Shneiderman, B., & Mayer, R., *Towards a cognitive model of programmer behavior*, Computer Science Technical Report No. 37, Indiana University, August 1975.