Memory & Cognition
1981, Vol. 9 (4), 422-433

Problem solving and the development of abstract categories in programming languages

BETH ADELSON
Harvard University, Cambridge, Massachusetts 02138

The question of how novice and expert computer programmers represent and use programming concepts is addressed here. Lines of programming code forming three complete programs were presented one at a time and in random order in a multitrial free recall procedure. Qualitative and quantitative measures revealed clear but different subjective organization in the two groups. The novices used a syntax-based organization, whereas the experts used a more abstract hierarchical organization based on principles of program function.

It is possible to look at a programming solution produced by an expert computer programmer and to be struck by the clever representation of the original problem, by the elegance of the style, and by the efficiency of the code. How is this finely crafted end product achieved? What are the skills of the expert, and how do they develop? These questions are addressed here by examining the conceptual structures of novice and expert programmers and investigating the principles underlying the conceptual structures.

Although there has been little theoretical research on computer programming, the literature on chess, go, and physics gives us relevant background information. De Groot (1965) found that master chess players of average memory capacity could reproduce more than 90% of a midgame board (20 pieces with correct item and order information) after only 5 sec of study. De Groot attributed their above-capacity performance to an ability to classify groups of pieces as instances of familiar playing categories. Unfortunately, de Groot did not actually identify the chunks formed by the masters, and so he could not study the nature of the chunks or how they were used. Chase and Simon (1973) replicated de Groot's findings and went on to isolate and then characterize the chess masters' chunks. They found that chunks frequently consisted of chess pieces that form attack or defense configurations. This suggests that the masters have identified the functional relationships that occur between the pieces during the game and that they have used these functional relationships to create internal representations of typical chess configurations.

This research was supported by Grant BNS-79-12418 from the National Science Foundation to Stephen M. Kosslyn. Thanks are given to Stephen M. Kosslyn and Carol Krumhansl, and also to Edward E. Smith, Endel Tulving, Roger Brown, and William K. Estes for editorial comments. Requests for reprints should be sent to Beth Adelson, Department of Psychology and Social Relations, Harvard University, Cambridge, Massachusetts 02138.

Chase and Simon also found that the chess master recalled a larger number of chunks than the chess expert. Because the number of chunks recalled by the master was above average but memory capacity was not, Chase and Simon suggested that the chunks were organized hierarchically. Each chunk was further chunked to form a configuration of a small number of larger, functionally related clusters. Although Chase and Simon had no direct evidence for this suggestion, the results of this experiment will provide evidence in accord with their idea.

Reitman (1976) found that master go players also encode game boards as functional clusters. She found, as did Chase and Simon (1973), that pieces forming attack or defense configurations were encoded together, although the go chunks seemed to form overlapping, rather than hierarchically organized, clusters. It would be interesting if the differences between go and chess made this a principled difference. Recently, several technical areas have also been the object of expert vs. novice skill research. Egan and Schwartz (1979) found that skilled electronic technicians also recall the elements of a circuit diagram in functional chunks. In addition, the technicians can rapidly identify a concept that serves to relate elements in a chunk and will systematically search circuit drawings for elements that are conceptually related. The chunking results of Egan and Schwartz lead us to infer that, through experience, the technicians have developed functionally based schemata. The search data suggest that the technicians have also developed knowledge about how to use these schemata. Chi, Feltovich, and Glaser (Note 1) have studied the behavior of experts and novices in solving physics problems. They suggest that both experts and novices form a schema that contains a description that has been obtained from the surface features of the problem. However, there are two differences between the schema of the novice and that of the expert. The schema of the novice is directly based on the surface features of the problem statement, whereas the schema of the expert

0090-502X/81/040422-12$01.45/0

is based on the physical principles underlying the problem statement. The expert schema also contains information about how and when to use each of the principles present in the schema. Chi et al. suggest how experts use information, as well as how they represent it. Although both questions are important, questions about the use of knowledge are addressed less frequently.

In their exploration of the concept of "physical intuition," D. P. Simon and H. A. Simon (1978) provide further evidence that experts form abstract functional descriptions of problem statements. Simon and Simon found their experts tended to solve problems with equations containing variables whose values were not given in the original problem statement. These values, however, could be derived from the original information, and so, the use of equations with these values suggests that an abstract and more useful representation of the problem was constructed by the experts in order to solve the problem.

Computer programming, like the four skill areas discussed above, provides another "semantically rich" domain (Bhaskar & Simon, 1977; H. A. Simon, 1979). The skills of the expert programmer are impressive. The representation and implementation of programming solutions can be nontrivial tasks requiring both linguistic and quantitative facility (Wolfe, 1977). Unfortunately, there has been very little theoretical research on computer programming; most research in this area has been on applied issues, such as how to identify good candidates for programming jobs. Because programming provides us with a large, natural, and highly structured problem space, it may prove to be theoretically as well as semantically rich.

There are three research questions to be asked here: How is the knowledge of the novice and the expert structured? What are the underlying principles used to structure that knowledge? And how is the information in those structures accessed and used? The experiment described in this paper was designed to study the knowledge structures of novice and expert programmers by looking at the proximities in their free recall (FR) data. The proximities between items in a subject's FR protocol give us evidence of his or her subjective organization (Friendly, 1977; Tulving, 1962). This, in turn, allows us to make inferences about the underlying cognitive representations that give rise to the observed organization. If a structured list of items is presented in random order by an experimenter, subjects will order their recall according to that nonrandom structure (Bousfield & Bousfield, cited in Crowder, 1976, p. 334). If a list of items is apparently unstructured, subjects will find or build a structure of their own; although the order of presentation changes from trial to trial, they will recall the list in a consistently ordered fashion (Sternberg & Tulving, 1977). Therefore, if a list is structured so that it contains more than one possible basis for organization, the organization of the subjects' responses gives us information about the conceptual structures the subjects are using during encoding and/or retrieval.

Although FR is not one of the usual tasks of a programmer, it is a good experimental tool for eliciting the structures built up through experience (Craik & Lockhart, 1972). For example, subjects recall randomly presented categorized lists by category. This is not because they are accustomed to having to recall the items together, but rather, because they have organized the items together over time. In addition, Reitman and Rueter (1980) suggest that, given an unusual task, subjects will very likely resort to using the structures and processing categories that they use most frequently. The recall tasks used in the expert-novice skill research summarized above produced sensible results; they did not seem to distort the subjects' behavior. In addition, Shneiderman (1976) found that level of recall was a good indicator of program comprehension, composition, and debugging skill. It was therefore believed that a recall task would be both useful and valid in the present situation.

In the experiment described here, novice and expert programmers were shown randomly ordered lines of Polymorphic Programming Language (PPL) code. The lines could be organized as belonging to one of the syntactic categories of the programming language. Syntax is being used here as it is used in the description of a natural language. Different control phrases of the programming language act as different parts of speech; that is, the function of the line itself and the types of lines expected to precede and/or follow it remain the same, regardless of the overall function of the program in which the phrase occurs. The lines could also be organized as belonging to one of the three programs that composed the stimulus set.

The organization found in the experts' and novices' recall protocols will give us information on the subjective organization of the two groups, information about what abstract principles of programming are used by novices and experts to structure their knowledge and how the structure of the knowledge thus facilitates and constrains its use. Although it is not possible to say whether the locus of the effect of the programmer's skill is at encoding or retrieval, it is possible to say that these conceptual structures are available and do influence performance.

METHOD

Subjects

The subjects formed two groups, novices and experts. The novice group consisted of five Harvard undergraduates who had just completed an introductory course in computer progamming in PPL. The expert group consisted of five teaching fellows for the same course. All subjects were paid volunteers.

Stimuli

The stimulus set consisted of 16 lines of PPL code. PPL was fresh in the minds of both groups of subjects, since the novices

Table 1
Stimulus Materials Labeled According to Each Item's Program Membership

1	2	
		Program 1
1	[1.0]	[x] $sort.within(sorting.list);i,j, temp.sort
2	[1.1]	[x] for i <– 1:length(sorting.list) dothru %3
3	[1.2]	[x] for j <–i:length (sorting.list) dothru %3
4	[1.3]	[x] if sorting.list[i]>sorting.list[j] then, temp.sort<–sorting.list [j]; sorting.list[j] <–sorting.list [i]; sorting.list[i] <–temp.sort
5	[1.4]	[x] return sorting.list
		Program 2
6	[2.0]	[x] $random.to.new(old.list);new.list,n,random.place
7	[2.1]	[x] new.list<– make(tuple,length(old.list),null);n<–0
8	[2.2]	[x] loop:n<–n+1;if n=length(old.list)+1 then return new.list
9	[2.3]	[x] random.place<–int(length(old.list)*random(0))+1
10	[2.4]	[x] if old.list[random.place] #null then new.list[n] <–old.list[random.place]; old.list[random.place] <– null; goto loop else goto %3
		Program 3
11	[3.0]	[x] $random.within(random.list);c,d, random.test, temp.within
12	[3.1]	[x] for c <– 1:length(random.list) dothru %4
13	[3.2]	[x] for d <– 1:length(random.list) dothru %4
14	[3.3]	[x] random.test <– int(2*random(0))+1
15	[3.4]	[x] if random.test=1 then temp.within <– random.list[d]; random.list[d] <– random.list[c]; random.list[c] <– temp.within
16	[3.5]	[x] return random.list

Note–Column 1 indicates the number of each item. The first number in Column 2 indicates the program in which the item occurs; the second number indicates its position within the program.

had just completed taking and the experts had just completed teaching a course in PPL. In addition, the underlying concepts used in the stimuli had all been presented during the semester to both groups of subjects. PPL is semantically similar to APL, having similar logical functions and data definition facilities, although PPL is more varied syntactically. PPL is similar to PL/I, both semantically and syntactically. PPL was developed to be a general language having properties in common with several other languages. This increases the possibility that the results of this experiment are general, rather than specific to PPL.

The heart of this experiment lies in the two possible bases of organization of the stimulus set, procedural and syntactic. Under the first basis (Table 1), the 16 lines can form three separate programs, with each line of code forming a single item. In Table 1, Column 1 presents the item number. In Column 2, the first number in the bracket is the number of the program that the item comes from. The second number is the position of the item in the program. (These numbers were not present when the subjects saw the items.) Items 1-5 (numbered [1.0] to [1.4] in Column 2) form a sorting routine which orders a list of items according to their original value. Successive pairwise comparisons are made to determine which items within the list will have their positions switched. The original list is returned once it is sorted. Items 6-10 (or [2.0] to [2.4]) form a routine that randomly orders a list of items by sampling an item from the original list randomly and without replacement and then placing the sampled item in a new list. Items 11-16 ([3.0] to [3.5]) form a routine that also randomly orders a list of items; however, its algorithm is similar to that of the sorting routine. Successive pairs of items switch positions within the list according to a randomly generated binary test. The original list is then returned with items randomized. Program 3, then, is similar to Program 1 in how it functions and similar to Program 2 in what it does, so Program 3 has procedural similarity to Program 1 and functional similarity to Program 2. Alternatively, each of the 16 lines falls into one of five syntactic categories. As mentioned earlier, membership in a syntactic category is determined by the control words of the line, which determine the line's function in the program. A line can be classified as a "function HEADER," which names the function and passes its arguments to it, as a "FOR statement," which controls an iteration process, as an "IF statement," which performs an operation if the conditions of the statement are met, as an "assignment statement," which assigns a value to a variable, or as a "RETURN statement," which allocates memory space for a variable. In Table 2, the stimulus set has been regrouped according to syntactic category, with Column 1 listing the item number and Column 2 listing the syntactic category of the item. Syntax is the second basis for organizing the stimulus set.

Each variable name occurs in only one of the three programs, and variable names were designed to be appropriate mnemonics for the program in which they occur.

Procedure and Apparatus

A multitrial FR (MTFR) procedure was used, with study and recall trials alternating. A subject saw a single line of code for 20 sec. The screen was then cleared and another line of code was shown, until all 16 lines had been viewed separately. Lines were presented without numbers. The order of presentation was random, so that any line from any of the three programs could follow any other line; that is, item presentation was not blocked either by routine or by syntactic classification. Subjects were then given 8 min to recall as many of the items as they could. This procedure was repeated for a total of nine trials. At the

Table 2
Stimulus Materials Labeled According to Each Item's Syntactic Category

1	2	
		Function Name or "HEADER" (H)
1	H	[x] $sort.within(sorting.list);i,j, temp.sort
6	H	[x] $random.to.new(old.list/new.list,n,random.place
11	H	[x] $random.within(random.list);c,d,random.test,temp.within
		Iteration or "FOR" Statements (F)
2	F	[x] for i <- 1:length(sorting.list) dothru %3
3	F	[x] for j <- 1:length(sorting.list) dothru %3
12	F	[x] for c <- 1:length(random.list) dothru %4
13	F	[x] for d <- 1:length(random.list) dothru %4
		Conditional or "IF" Statements (I)
4	I	[x] if sorting.list[i] >sorting.list[j] then; temp.sort <- sorting.list[j]; sorting.list[j] <- sorting.list[i]; sorting.list[i] <- temp.sort
8	I	[x] loop:n <- n+1;if n=length(old.list)+1 then return new.list

10	I	[x] if old.list[random.place] #null then;
		new.list[n] <- old.list[random.place];
		old.list[random.place] <- null;goto loop else goto %3

15	I	[x] if random.test=1 then temp.within <- random.list[d]; random.list[d] <- random.list[c]; random.list[c] <- temp.within
		Assignment Statements (A)
7	A	[x] new.list <- make(tuple,length(old.list),null);n <- 0
9	A	[x] random.place <- int(length(old.list)*random(0))+1
14	A	[x] random.test <- int(2*random(0))+1
		Return Statements (R)
5	R	[x] return sorting.list
16	R	[x] return random.list

Note–Column 1 indicates the item's number; Column 2 indicates the item's syntactic category.

beginning of the experiment, subjects were told what their task would be; they were told that the stimulus materials could be organized in a number of ways to form a cohesive whole. They were told that it might help them to organize the material, but they were also told that, across trials, they were free to either change or keep any organization that they had chosen. Therefore, both bases of organization were potentially available to both groups of subjects.

A PPL program was written by the author to run the experiment (Adelson, Note 2). The instructions and stimulus materials were presented on the screen of a terminal that was linked to a PDP-11/70 computer. Subjects typed their responses at the terminal, and their interresponse times were recorded. The responses were cleared from the screen at the end of the trial. The experiment lasted 2 h.

RESULTS AND DISCUSSION

An analysis of variance was performed on the number of items recalled. The experts recalled more than the novices [mean = 12.733 > mean = 9.600; $F(1,8) = 12.38$, $p < .01$], and this difference remained constant across trials [$F(1,8) = 1.78$, $p > .05$]. The pause before each response was recorded, and an item recalled with less than a 10-sec pause preceding it was considered a member of the current chunk. Using this criterion, chunks are groups of items recalled successively and continuously (Chase & Simon, 1972). An analysis of variance was performed on chunk size, with chunk size being measured by the number of items in a burst of recall. Chunk size was greater for experts than for novices [mean = 3.48 > mean = 2.37; $F(1,8) = 68.67$, $p < .01$], with the difference remaining constant over trials [$F(1,8) = 2.61$, $p > .05$]. The consistency of a subject's recall order across trials under FR conditions is regarded as evidence of subjective organization (SO). A measure of SO was computed, using Sternberg and Tulving's (1977) measure of pair frequency. $PF = O(ITR2) - E(ITR2) = O(ITR2) - 2C(C-1)/hk$, where PF represents pair frequency. Pair frequency measures the observed number of pairs of items recalled together (in any order) on a pair of successive trials, O(ITR2), minus the number of item pairs that would be expected to occur together on a pair of trials due to chance, [E(ITR2)]; C = the number of items recalled on both trials, h = the number of items recalled on the first pair of trials, and k = the number of items recalled on the second of a pair of trials. An analysis of variance was done of the pair frequency measure for the two groups averaged over pairs of trials. The pair frequency of the experts was found to be greater than the pair frequency of the beginners [mean = 8.21 > mean = 1.79; $F(1,8) = 68.67$, $p < .01$], and the advantage of the experts increased slightly across trials [$F(1,8) = 10.05$, $p < .05$].

Comparing each group's pair frequency score with

its maximum possible pair frequency score allows us to then compare their subjective organizations while taking into account their differing levels of recall. If both groups had been completely consistent in their recall orders across all pairs of trials, then on Trial t + 1, regardless of how many new items they recalled, the subjects would have still recalled all of the old items that they had recalled on Trial t together and in the same order. The pair frequency score for the experts would then have been 9.75, compared with the 8.75 obtained, and the pair frequency score for the novices would have been 6.93, compared with the 1.79 obtained. The experts' pair frequency score was 84.0% of their maximum, and that of the novices was 25.8% of their maximum, suggesting that the subjective organization of the experts was stronger.

It is not surprising that the experts recalled more than the novices, had larger recall chunks, and had more consistent subjective organization (as evidenced by their higher pair frequency scores). There was also greater similarity of recall order among the experts than among the novices. This is suggested by the two-dimensional scaling solution of intersubject recall similarity that is presented in Figure 1.

To obtain this solution and all of the subsequent scaling and clustering solutions, a distance matrix was constructed for each subject on each trial. In these matrices, distance was measured by counting the number of items intervening between any two recalled items. These raw distance matrices provided a measure of interitem similarity for each subject on each trial. The matrices could then be averaged over subjects and/or trials by averaging the distances in each cell in the matrix. Friendly's (1977) formula was used to compute the distances obtained here.

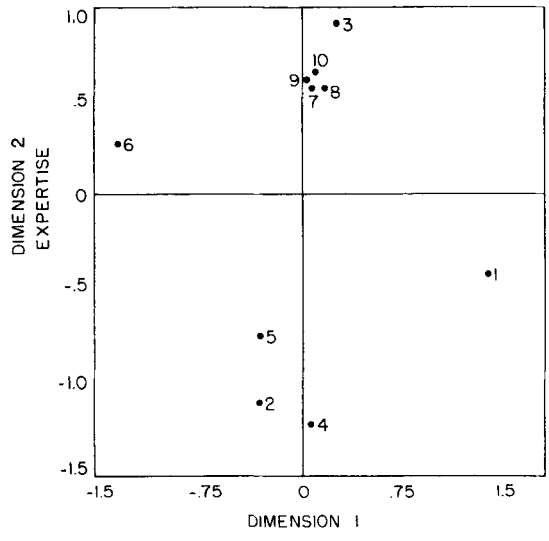


Figure 1. Two-dimensional scaling solution for intersubject recall similarity (Subjects 1-5 are novices; Subjects 6-10 are experts).

To obtain the solution for intersubject recall similarity, an intersubject correlation matrix was constructed. This matrix was constructed by averaging over the distance matrix for each subject on each recall trial, yielding one matrix per subject averaged over trials. These matrices were then correlated, yielding the intersubject similarity matrix. Scaling solutions were then obtained for this new matrix in one, two, and three dimensions using the Kyst program (Kruskal, Young, & Seery, Note 3). When stress was plotted as a function of dimensionality, the decrease in stress was greatest when dimensionality increased from one to two. Kruskal and Wish (1978) suggest that this decrease may be used in combination with the interpretability of the solutions as a guide in choosing among solutions having differing numbers of dimensions. The two-dimensional solution, which yielded the greatest decrease in stress and produced interpretable clusters, is the solution presented here. Stress value was reasonable (Stress 1 = .080); inspection of the scatter plot indicated that the solution was not degenerate, and a second run of the program yielded similar results, suggesting that a local minimum was not reached.

With one exception (Subject 6), the expert subjects (Subjects 6-10) are clustered closely together near the zero point on Dimension 1 and above the zero point on Dimension 2. The novice subjects (Subjects 1-5) cluster less closely with each other and, with the exception of Subject 3, fall below the zero point on Dimension 2. This separation of subjects suggests that Dimension 2 represents expertise. It is more difficult to name Dimension 1, which therefore has not been labeled. The greater similarity among the experts than among the novices suggests that the more skilled subjects have a stronger conceptual basis for interpretation and comprehension that they bring to bear even in unusual circumstances. It is interesting that the variability among subjects decreased with skill. It is possible that the expert subjects are similar in their basic processes, such as comprehension, or are similar when processing materials that are not too complex in nature. Perhaps a more open-ended task or more complex materials would reveal individual differences on higher level skills.

It is interesting now to look at the scaling and clustering solutions of interitem similarity for the novice and expert groups. The qualitative analyses give us information about the conceptual representations of each of the groups and about the principles underlying each group's representation. They also suggest how these principles are used and, so, let us make inferences as to why we find the quantitative differences described above.

Scaling solutions were obtained for the novice and expert groups separately in one, two, and three dimensions. For both groups, when stress was plotted as a function of dimensionality, the decrease in stress was greatest when dimensionality increased from one to two. The two-dimensional solution also yielded the most

clearly interpretable clusters. As a result, the two-dimensional solutions are presented here.

The two-dimensional scaling solution for the expert group is shown in Figure 2. Reasonable stress was obtained (Stress 1 = .0218). Examination of the scatter diagram suggested that the solution was not degenerate, and a second and third run of the program both yielded similar results, suggesting that a local minimum was not reached. In Figure 2, we see that for the experts, items cluster according to program membership. The clusters are fairly compact and quite distinct; Program 1 clusters in the upper right-hand quadrant, Program 2 in the lower left, and Program 3 in the lower right. This indicates that the expert subjects used program membership as a basis for the inclusion and exclusion of items. Dimension 1 is labeled new vs. old variable returned because Programs 1 and 3, which return the variables originally passed to them, both fall on the high end, whereas Program 2, which returns a new variable, falls on the low end. Dimension 2 has been labeled random vs. ordered because the two randomization programs fall at the low end of the dimension, whereas the sorting routine falls at the high end.

The two-dimensional scaling for the novice group is shown in Figure 3. Reasonable stress was obtained (Stress 1 = .1825). Examination of the scatter plot

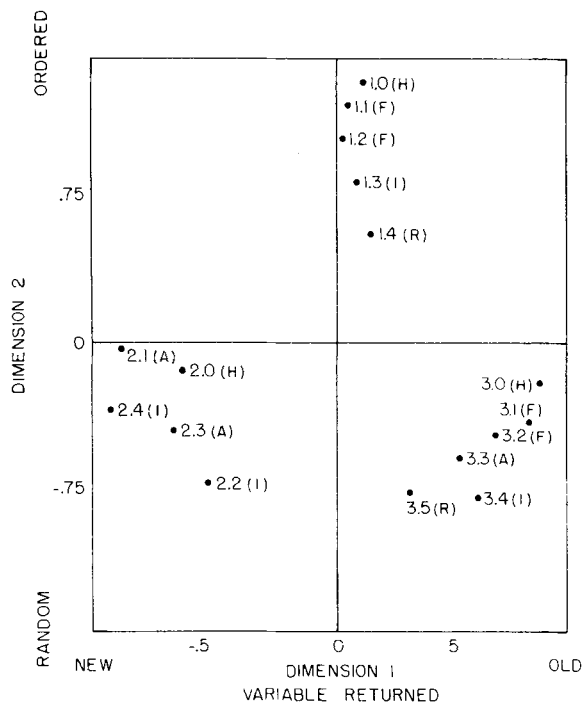


Figure 2. Two-dimensional scaling solution for interitem similarity of expert group on Trials 6-9. Points are numbered by item number: The first number indicates program membership; the second number indicates position in the program. (Syntactic classification appears in parentheses.)

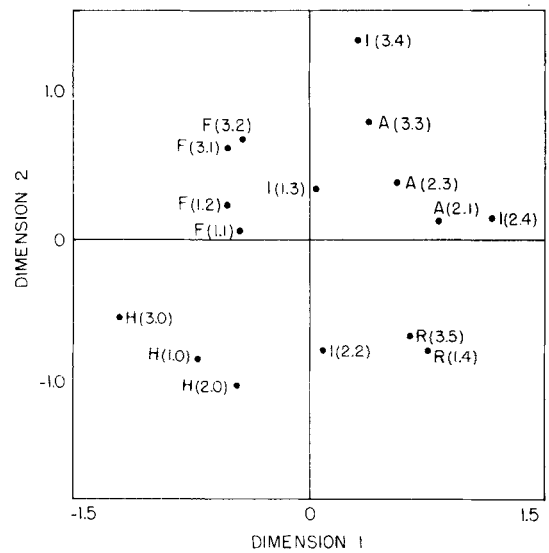


Figure 3. Two-dimensional scaling solution for interitem similarity of novice group on Trials 6-9. Letters indicate the syntactic category of each point. (Program membership appears in parentheses.)

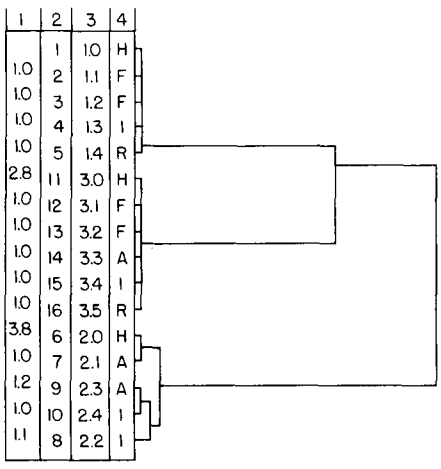
suggested that the solution was not degenerate, and second and third runs of the program yielded similar results, suggesting that a local minimum was not reached. We see in Figure 3 that the iteration or FOR statements (labeled F) cluster in the upper left quadrant. The function HEADERS, or names (labeled H), are in the lower left; the RETURN statements (labeled R) are in the lower right, and the conditional IF statements (labeled I) and the assignment statements (labeled A) are both in the upper right quadrant. The alternative classification of each item (i.e., its program membership) is given in parentheses. For the novice group, the items cluster according to syntactic category and not according to program membership, and so we can infer that syntactic category is an organizing principle at this level of expertise.

The scaling solutions give us an overall picture of the organization for each of the two groups. We see that the novices have clusters of items from the same syntactic category, whereas the experts have clusters of items belonging to the same program. It is possible to supplement the information from the hierarchical clustering solutions. In the ideal case, this would not be appropriate. There, one of the solutions would provide a perfect model of the data, and then the other solution would be inaccurate. However, as long as neither model provides a perfect fit and both models are found to satisfy reasonably strict goodness-of-fit criteria, it does seem appropriate to use the information in both. This is especially true here, since the information in the two solutions is complementary (Kruskal, 1977; Kruskal & Wish, 1978). The scaling solution provides global information regarding what items make up a cluster, and the

clustering solution provides information about the organization of items within a cluster.

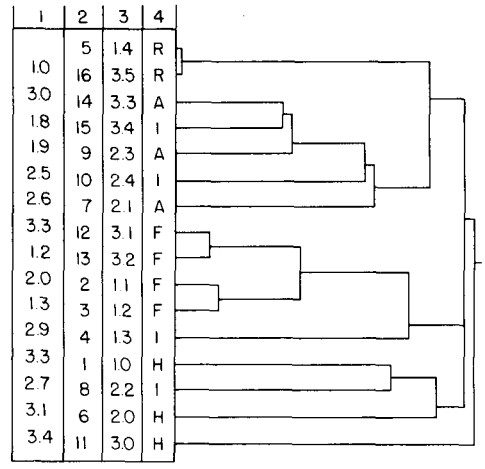
Using the program Agclus (Oliver, Note 4), a hierarchical cluster analysis was computed on the two distance matrices obtained by averaging over Trials 6-9 for each subject and then by averaging over each of the subjects in the novice and expert groups. Friendly (1977) suggests that goodness of fit can be evaluated for a hierarchical cluster analysis by comparing the solutions obtained under the program's minimum cluster diameter and maximum cluster diameter clustering criteria. Under the minimum method, clusters (or items) are joined if any member of the first cluster is maximally close to any member of the second cluster. Under the maximum method, all members of both clusters must be closer to each other than to any other item before the two clusters can be joined. The program was run twice, once under the minimum criterion and once under the maximum criterion. Both criteria yielded similar results for each group, suggesting that the model's violation of the ultrametric assumption is minimal.

Figures 4 and 5 present the solutions for the experts and the novices, respectively. The solutions shown are only for the minimum diameter criterion, since they were quite similar to the maximum solutions. A general description of how to read the cluster trees presented in Figures 4 and 5 follows. The trees are read from left to right. Following the branch from an item on the left out to a node on the right allows you to see which items are clustering together. Branch length represents the distance between an item and its closest neighbor. The length of the longest branch from a group of clustered items to a node represents the diameter of the cluster containing those items. Clusters can also be joined with other clusters or with other items into larger clusters. These clusters with larger diameters have their nodes further to the left on a given tree (since larger diameters are represented by longer distances from items to nodes). The cluster trees are derived from the interitem proximities found in the subjects' recall data. Because these proximities reflect the relationships present in the subjects' internal representations of the items, the trees provide models of the subjects' representations. The relationship between the data and the trees will be discussed shortly.



1	2	3	4
	1	1.0	H
1.0	2	1.1	F
1.0	3	1.2	F
1.0	4	1.3	I
1.0	5	1.4	R
2.8	11	3.0	H
1.0	12	3.1	F
1.0	13	3.2	F
1.0	14	3.3	A
1.0	15	3.4	I
1.0	16	3.5	R
3.8	6	2.0	H
1.0	7	2.1	A
1.2	9	2.3	A
1.0	10	2.4	I
1.1	8	2.2	I

Figure 4. Hierarchical clustering for the expert group (Trials 6-9) (Column 1 = cluster diameter; Column 2 = item number; Column 3 = program membership; Column 4 = syntactic category.)



1	2	3	4
	5	1.4	R
1.0	16	3.5	R
3.0	14	3.3	A
1.8	15	3.4	I
1.9	9	2.3	A
2.5	10	2.4	I
2.6	7	2.1	A
3.3	12	3.1	F
1.2	13	3.2	F
2.0	2	1.1	F
1.3	3	1.2	F
2.9	4	1.3	I
3.3	1	1.0	H
2.7	8	2.2	I
3.1	6	2.0	H
3.4	11	3.0	H

Figure 5. Hierarchical clustering for the novice group (Trials 6-9) (Column 1 = cluster diameter; Column 2 = item number; Column 3 = program membership; Column 4 = syntactic category.)

To read the clustering solution for the experts (Figure 4), look at the branches from the first five items in Column 3 (the items classified according to program membership as 1.0, 1.1, 1.2, 1.3, and 1.4). They are all of Length 1, and so the cluster diameter of the five items composing Program 1 equals one (cluster diameters appear in Column 1, to aid visual inspection). Continuing with the clustering solution for the experts, we see from the length of their branches that the items from Program 3 are also contained in a cluster with a diameter of one. The items from Program 2 cluster with a diameter of 1.2. In Figure 4, we see that, with the exception of Items 2.2 and 2.3, cluster diameter does not change as new items from a given program are added to a cluster. This indicates that the distances between each pair of items are equal, as they would be in a set of serially related items, such as the lines of a computer program.

The serial relationship of items within the clusters, as indicated by the equal distance found between the items, occurs in the clustering solution because in the actual recall protocols, the second line of Program 1 is recalled right after the first line of Program 1 and the third line of Program 1 is recalled right after the second line, and so on (despite the fact that a different random presentation order was used for each trial). This can be seen by looking at the actual recall data, which are presented in Table 3.

Table 3
Recall Data for the Expert Group Averaged Over Trials 6-9

	Subject				
OP	6	7	8	9	10
1	3.0 (H)	1.0 (H)	1.0 (H)	1.0 (H)	1.0 (H)
2	3.1 (F)	1.1 (F)	1.1 (F)	1.1 (F)	1.1 (F)
3	3.2 (F)	1.2 (F)	1.2 (F)	1.2 (F)	1.2 (F)
4	3.3 (A)	1.3 (I)	1.3 (I)	1.3 (I)	1.3 (I)
5	3.4 (I)	1.4 (R)	1.4 (R)	1.4 (R)	1.4 (R)
6	3.5 (R)	3.0 (H)	3.0 (H)	3.0 (H)	3.0 (H)
7	1.0 (H)	3.1 (F)	3.1 (F)	3.1 (F)	3.1 (F)
8	1.1 (F)	3.2 (F)	3.2 (F)	3.2 (F)	3.2 (F)
9	1.2 (F)	3.3 (A)	3.3 (A)	3.3 (A)	3.3 (A)
10	1.3(I)	3.4 (I)	3.4 (I)	3.4 (I)	3.4 (I)
11	1.4(R)	3.5 (R)	3.5 (R)	2.0 (R)	3.5 (R)
12	2.3(A)	2.0 (H)	2.0 (H)	2.1 (A)	2.0 (H)
13	2.4 (I)	2.1 (A)	2.1 (A)	2.2 (I)	2.1 (A)
14	2.0 (H)	2.2 (I)	2.2 (I)	2.3 (A)	2.2 (I)
15	2.1 (A)	2.3 (A)	2.3 (A)	2.4 (I)	2.3 (A)
16		2.4 (I)	2.4 (I)		2.4 (I)

Note–OP = ordinal position of items. Each column presents the recall data for one subject. The ordering of the items represents the order in which they were recalled, averaged over Trials 6-9. Items are labeled according to program membership; the syntactic category of each item appears in parentheses.

Here, the recall data are presented for each expert subject, averaged over Trials 6-9. Items occur in each column in the order in which they were recalled. They are labeled according to their program membership. The syntactic category of each item is given in parentheses. Reading down each column, we see that every subject recalled all the items from a given program together. In addition, except for Subject 6, who inverted the first and second halves of Program 2, all of the items were recalled in the order in which they would be evaluated during the execution of the program. The order in the data is striking and clearly supports the validity of the clustering solution. The clusters that appear both in the data and in the hierarchical clustering indicate that the expert subjects used their knowledge of the serial nature of program execution to organize items within categories. This serial relationship was suggested in the two-dimensional scaling solution. In Figure 2, we saw that the items from Programs 1 and 3 seemed to fall along Dimension 2 according to their serial positions in the program; however, the relationship emerges most clearly in the hierarchical clustering.

The data from the scaling and clustering solutions indicate that the experts perceive each line as part of a functional whole: Individual lines are no longer separate entities to be grouped according to syntactic category; they have become parts of a more abstract, more conceptually complex entity based on the subject's knowledge of computational procedures.

A second level of organization is present in the experts' clustering solution. Recall that Program 1 and Program 3 are similar in the computational procedure used. The clusters containing the items of Programs 1 and 3 are the next clusters to join in the clustering solution. This can be seen by following the branches from those two clusters out to the node that joins them at a distance of 2.8. This reflects the groupings in the data in Table 3. Reading down the columns, we can see that Program 1 and Program 3 were recalled contiguously by all of the expert subjects. The exclusive nature of this clustering program, in which clusters are joined to one cluster or another and any cluster overlap is not represented, does not let us see any possible effect of the similarity of function shared by Programs 2 and 3. However, this representation does suggest that the conceptual structure of the experts is hierarchical in nature. This is particularly interesting because Chase and Simon (1973) stated that the greater number of chunks recalled by their master than by their Class A player suggested the existence of a hierarchical relationship between chunks for the master.

Figure 6 shows the clustering solution embedded in the scaling solution. The solid lines surround the three clusters whose diameters are 1.2 or less. The broken line surrounds the cluster whose diameter is 2.8. This combined solution underscores the interitem and intercluster relationships present in the data. It clearly illustrates the clustering of items according to their

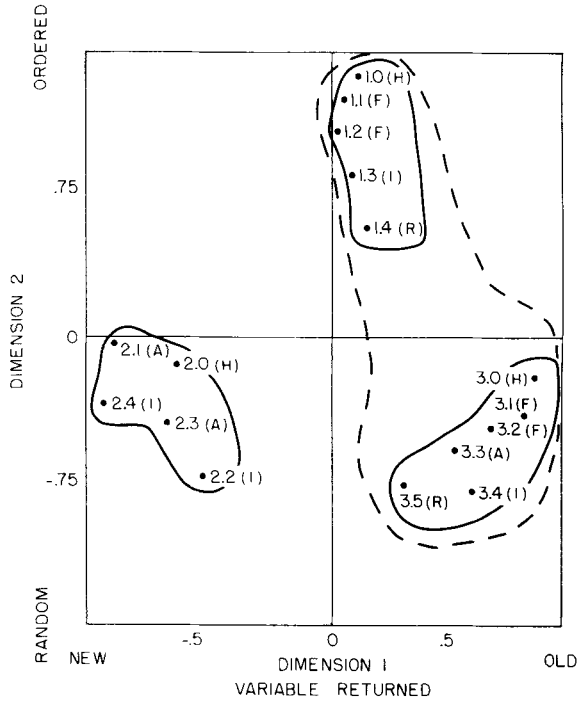


Figure 6. The hierarchical clustering embedded in the two-dimensional scaling solution for the expert group for interitem similarity. The solid lines surround clusters of diameter 1.2. The broken line surrounds the cluster of diameter 2.8. Numbers indicate the program membership of each item. The syntactic category appears in parentheses.

program membership and the further clustering of programs according to their procedural similarity.

Turning to the cluster analysis for the novice group (Figure 5), we see a very different within-category organization. Reading down Column 4, we first find the two RETURN statements (labeled R) clustering with a diameter of 1.0 (as given in column 1). Next, the conditional, or IF, statements (labeled I) and assignment statements (labeled A) cluster at a diameter of 2.6; then the iteration, or FOR, statements (labeled F) and, finally, the function titles or HEADERS (labeled H) cluster with the rest of the conditional IF statements (labeled I). In this solution, as in the multidimensional scaling solution, the items do cluster by syntactic category, and once again, these clusterings reflect the groupings found in the actual recall data.

In Table 4, the recall data are presented for each novice subject, averaged over Trials 6-9. Items occur in each column in the order in which they were recalled. They are labeled according to their syntactic category, and the program membership of each item is given in parentheses. Reading down each column, we see that the items from each syntactic category seem to be recalled together. For example, Subject 4 recalls the RETURN (R) statements, then the function HEADERS (H), one IF (I), one FOR (F), then two assignment statements (A), the three other FORs (F), and then one IF (I), and one last assignment statement (A). Not surprisingly, the data for the novices show more variability than the data for the experts. However, the syntactic groupings are present and do support the clustering solution. The presence of these syntactic groupings is further supported by the results of the proximity analysis that is presented shortly.

Table 4
Recall Data for the Novice Group Averaged Over Trials 6-9

	Subject				
PO	1	2	3	4	5
1	A (2.3)	I (2.2)	H (3.0)	R (1.4)	H (3.0)
2	A (3.3)	I (1.3)	F (1.1)	R (3.5)	H (1.0)
3	I (2.4)	A (2.3)	F (3.1)	H (3.0)	H (2.0)
4	I (1.3)	A (2.1)	F (1.2)	H (2.0)	I (2.2)
5	I (2.2)	R (3.5)	F (3.2)	H (1.0)	F (1.1)
6	R (1.4)	R (1.4)	I (3.4)	I (1.3)	F (1.2)
7	R (3.5)	A (3.3)	R (3.5)	F (1.1)	F (3.1)
8	I (2.2)	H (2.0)	R (1.4)	A (2.1)	I (1.3)
9	F (3.1)	H (1.0)	A (3.3)	A (2.3)	F (3.2)
10	F (3.2)	F (1.2)	I (2.2)	F (1.2)	R (3.5)
11	H (2.0)	F (3.1)	A (2.3)	F (3.2)	R (1.4)
12	H (3.0)	F (1.1)	I (2.4)	F (3.1)	A (2.3)
13	F (1.1)	F (3.2)	I (1.3)	I (2.2)	A (3.3)
14	F (1.2)	I (3.4)	H (2.0)	A (2.3)	I (2.4)
15	A (2.1)		A (2.1)		I (3.4)
16	H (1.0)				A (2.1)

Note–OP = ordinal position of items. Each column presents the recall data for one subject. The ordering of the items represents the order in which they recalled, averaged over Trials 6-9. Items are labeled according to syntactic category; the program membership of each item appears in parentheses.

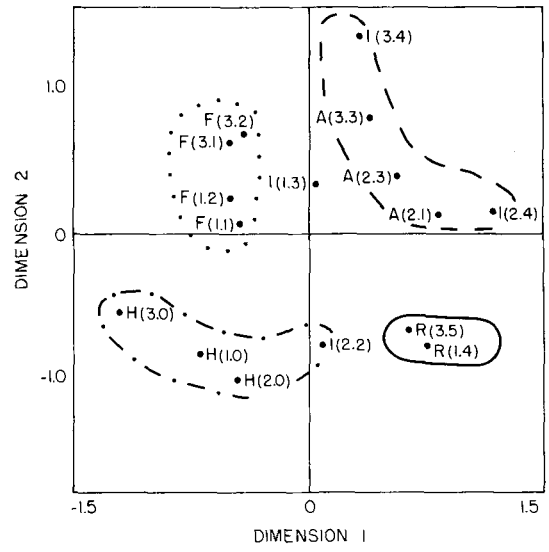


Figure 7. The hierarchical clustering embedded in the two-dimensional scaling solution for the novice group for interitem similarity. The solid line surrounds the cluster of diameter 1.0; the dotted line surrounds the cluster of diameter 2.0. The dashed line surrounds the cluster of diameter 2.6, and the dashed and dotted line surrounds the cluster of diameter 3.1. Letters indicate the syntactic category of each item. The program membership of each item appears in parentheses.

In Figure 7, the clustering is embedded in the scaling solution. The figure illustrates the clear separation of items according to syntactic category. The solid line around the Rs indicates their cluster diameter of 1.0. The dotted lines around the Fs indicates their diameter of 2.0. The dashed line around the As and Is indicates their diameter of 2.6, and the dashed and dotted line around the Hs indicates their diameter of 3.1.

In Figure 5, the clustering of the novice group, then, is very different from that of the expert group. Although clusters are present for the novices, they are syntactic rather than semantic and they are less visually striking. They are less striking because the distance between maximally close pairs of items within the cluster is more variable for the novices than for the experts. The variability of within-cluster pairwise distances suggests that, although the items are grouped by syntactic category, they are not arranged according to a clear organizing principle within the syntactic categories. That is, some items in a category may be near each other and others farther apart, but no discernible pattern emerges to suggest why this is the case.

Proximity of clustering (PC) is a measure of how well each classification (syntactic or procedural) describes the actual clustering proximities. Under the PC measure, a given hypothetical classification perfectly describes the clusterings found in the data if all of the items from each of the hypothesized categories are recalled contiguously in any order. In this case, the PC score for the hypothesized classification will be equal to 1. On the other hand, whenever an item from outside a hypoth-

esized category is recalled between two items belonging to the hypothesized category, the classification does not perfectly describe the data and the PC measure decreases. In the present case, classification of items by program membership perfectly describes the data if all of the items from each of the three programs have been recalled together, in which case the PC score will equal 1 for the program membership classification. If the clusters are not wholly discrete, the PC measure will decrease. The syntactic classification perfectly describes the data if all of the items from each syntactic category appear together; in this case, the PC score for the syntactic classification will equal 1. If it is not completely accurate, then items from other syntactic categories will occur between items from a single category and the PC measure will reflect this by decreasing. When a PC score is obtained for two competing classifications, the one that yields the higher PC score is the one that is a more accurate description. In order to quantify how well each classification described each group of subjects, PC measures were obtained for each group under each classification. PC was computed for each group, averaging over Trials 6-9, using Formula 1:

$$C = 1 - [(1/tm) \sum_{t} \sum_{m} [(1/NC_{tm})NI_{tm}]] \qquad (1)$$

where m = the total number of categories for a given classification, t = the number of trials, NC = the total number of items correctly recalled that do not belong to the category being considered, and NI = the total number of noncategory items correctly recalled between the first and last items of the category being considered. In this formula, PC decreases every time any item from outside a category occurs between the two most distant items of a given category. This is the meaning of NI_{tm} and this expression is equal to the number of noncategory items occurring between the two most distant items of a category. This is divided by NC_{tm} to equate for different levels of recall and then averaged over the number of categories in the classification and over trials. An analysis of variance was performed on the PC scores obtained for each group under each classification.

A significant interaction was found showing that the novice data were better characterized by the syntactic classification than by the procedural classification, and the expert data were better characterized by the procedural classification than by the syntactic [mean = .80 > mean = .53; mean = .99 > mean = .53; $F(1,8) = 60.93$, $p < .01$]. The main effect of group [$F(1,8) = 6.94$; $p < .05$] but not of classification [$F(1,8) = 5.00$, $p > .05$] was significant. The effects of category size and level of recall were balanced here, since the levels of group are crossed with the levels of classification. The significant interaction of group with classification further supported the characterization of the experts' clusters as procedural and the novices' clusters as syntactic.

In order to confirm the criteria used in classifying each subject as a novice or an expert and also to look for differences among novices who had different amounts of programming experience, several of the subjects were regrouped and their recall and pair frequency scores were reanalyzed. The three novices having the least experience (i.e., those currently enrolled in their first programming class) were grouped together and compared with the two novices having the most experience (those having completed more than two programming courses) plus the expert with the least experience. The results of two analyses of variance showed no significant difference in number recalled ($p > .05$) or in amount of subjective organization ($p > .05$). In addition, all subjects in the original novice group obtained higher clustering proximity scores for the syntactic classification than for the functional classification, whereas the reverse was true for all subjects in the original expert group. These results do not give us information as to what is the level of expertise at which the novice subjects reorganize their knowledge structures into the knowledge structures of the experts.

CONCLUSIONS

The data suggest that both experts and novices have conceptual categories for the elements of a programming language. For the novices, the categories seem to be syntactic rather than semantic in nature and the items within the category are not related to each other in a strongly organized manner. As expertise increases, the nature of the categories changes. They become more conceptually complex, changing from one line operations to entire routines, and they shift from being syntactically based to being semantically based. In addition, as suggested by the results of the cluster analysis, the first-level functional clusters are hierarchically organized at a second level of abstraction according to procedural similarity.

There are several interesting questions that arise at this point. There is substantial evidence that a basic level of conceptualization exists (Rosch & Mervis, 1975; Rosch, Mervis, Gray, Johnson, & Bayes-Braem, 1976) and is used to recognize concrete objects. For example, when subjects are shown pictures of objects that they are to name, they use the basic-level name. Delicious apples are called apples (the basic-level name) rather than delicious apples (the subordinate-level name) or fruit (the superordinate-level name). Kitchen chairs are called chairs, rather than kitchen chairs or furniture. In addition, subjects can verify that an item is an apple more quickly than they can verify that it is a delicious apple or a fruit. Rosch's results suggest that when objects are first presented, they are encoded as members of a basic-level category. A basic-level encoding provides

the subject with a representation that has moderately high numbers of functional and perceptual features (Rosch et al., 1976). This allows the subject to make use of the object while maintaining cognitive economy.

It is possible that a basic level also exists for computational concepts. The hierarchical cluster analysis suggested that the expert subjects chunked individual items into integral wholes that were then organized hierarchically according to procedural similarity. This hierarchical chunking suggests an underlying categorical encoding in which certain features are used as a basis for similarity and others are ignored. It is possible that these encoded concepts have the properties of the basic level. At least initially, they resemble Rosch's basic-level objects. Further research has been planned on how these concepts fare in situations that call for detailed vs. general understanding. This will give us information about whether these concepts do have the properties of the basic level, whether they really are the expert's most comfortable level of conceptualization, and how flexibly they can be used.

It would be interesting to trace the changes that occur with level of expertise. The data have provided no indication of a gradual progression and no indication of when the changes take place, since differences in the number of programming courses taken by the members of the novice group did not cause any of the novices to perform differently from each other or similarly to the subjects in the expert group.

Although there are methodological differences between the current study and the Chase and Simon (1973) chess study, the data of the two studies produce similar results. In both of these studies and in skilled problem solving studies in general, the experts have developed structures based on the functional principles of their area of expertise. It is possible that other similarities exist across problem solving domains. There is an apparent difference between the results of the present experiment and the results of the Chase and Simon experiment, but this difference disappears on closer inspection. Chase and Simon found that chess masters and novices had equal levels of recall when shown random, rather than actual, game boards. In the present experiment, experts had greater recall than novices when shown randomly ordered lines. However, the randomly ordered lines of our experiment still could be unscrambled to form a meaningful set and are therefore not exactly analogous to the random chess boards, which did not contain the same potential meaningfulness. The real analogy to the Chase and Simon random condition would have been 16 lines of unrelated code. If the Chase and Simon paradigm had been used here, novices and experts both would have been shown full programs in their proper order as well as unrelated lines of code in a random order. This procedure was considered, but it was decided that the different skills of the two groups of subjects would have a chance to emerge more clearly in a MTFR task in which one organization strategy or another was not imposed by the experimenter.

A last point is on the relationship between the comprehension and use of programming languages and natural languages. Are there "natural computational concepts"? Do existing programming languages capture them? Which semantic and syntactic forms of existing programming languages are in accord with our natural language skills? Which present the most difficulty? And what level of conceptualization is most useful at different levels of expertise? Further research in this area could be useful in the development of user-oriented programming languages.

REFERENCE NOTES

1. Chi, M., Feltovich, P., & Glaser, R. *Representation of physics knowledge by novices and experts* (Tech. Rep.). Pittsburgh, Penn: University of Pittsburgh, 1977.
2. Adelson, B. *A flexible program for running multi- and single-trial free recall experiments.* Manuscript in preparation, 1981.
3. Kruskal, J. B., Young, F. W., & Seery, J. B. *How to use KYST, a very flexible program to do multi-dimensional scaling and unfolding.* Murray Hill, N.J: Bell Telephone Laboratories, undated.
4. Oliver, D. *Using Agclus: A program to do hierarchical clustering analysis.* Unpublished manuscript, Harvard University, 1974.

REFERENCES

Bhaskar, R., & Simon, H. A. Problem solving in semantically rich domains: An example from engineering thermodynamics. *Cognitive Science,* 1977, **1**, 193-215.

Chase, W. C., & Simon, H. A. Perception in chess. *Cognitive Psychology,* 1973, **4**, 55-81.

Craik, F. I. M., & Lockhart, R. C. Levels of processing: A framework for memory research. *Journal of Verbal Learning and Verbal Behavior,* 1972, **11**, 671-684.

Crowder, R. G. *Principles of learning and memory.* Hillsdale, N.J: Erlbaum, 1976.

de Groot, A. D. *Thought and choice in chess.* The Hague: Mouton, 1965.

Egan, D. E., & Schwartz, B. J. Chunking in recall of symbolic drawings. *Memory & Cognition,* 1979, **7**, 149-158.

Friendly, M. L. In search of the M-gram: The structure of organization in free recall. *Cognitive Psychology,* 1977, **9**, 188-249.

Kruskal, J. B. The relationship between multi-dimensional scaling and clustering. In Van Ryzin (Ed.), *Classification and clustering.* New York: Academic Press, 1977.

Kruskal, J. B., & Wish, M. *Multidimensional scaling.* Beverly Hills, Calif: Sage, 1978.

Reitman, J. S. Skilled perception in go: Deducing memory structures from inter-response times. *Cognitive Psychology,* 1976, **8**, 336-356.

Reitman, J. S., & Rueter, H. Organization revealed by recall orders and confirmed by pauses. *Cognitive Psychology,* 1980, **12**, 559-581.

Rosch, E., & Mervis, C. B. Family resemblances: Studies in the internal structure of categories. *Cognitive Psychology,* 1975, **7**, 573-605.

Rosch, E., Mervis, C. B., Gray, W. D., Johnson, D. M., &

BOYES-BRAEM, P. Basic objects in natural categories. *Cognitive Psychology,* 1976, **8**, 382-439.

SHNEIDERMAN, B. Exploratory experiments in programmer behavior. *International Journal of Computer and Information Sciences,* 1976, **5**, 123-354.

SIMON, D. P., & SIMON, H. A. Individual differences in solving physics problems. In R. S. Siegler (Ed.), *Children's thinking: What develops?* Hillsdale, N.J: Erlbaum, 1978.

SIMON, H. A. Information processing models of cognition. *Annual Review of Psychology,* 1979, **30**, 383-396.

STERNBERG, R. J., & TULVING, E. The measurement of subjective organization in free recall. *Psychological Bulletin,* 1977, **84**, 539-556.

TULVING, E. Subjective organization in the free recall of "unrelated" words. *Psychological Review,* 1962, **69**, 344-354.

WOLFE, J. M. An interim validity report on the Wolfe Programming Aptitude Test. *Computer Personnel,* 1977, **6**, 1-2.

(Received for publication September 8, 1980;
revision accepted February 13, 1981.)