The Understanding Process: Problem Isomorphs

Herbert A. Simon and John R. Hayes
Carnegie-Mellon University

In previous papers (Hayes & Simon, 1974, 1975) the authors have developed a theory about human understanding of written instructions. The formal theory is embodied in a computer program, UNDERSTAND, which simulates these understanding processes. UNDERSTAND takes written problem instructions in natural language as input, and produces internal representations of the problems and the legal move operators as outputs. A program that is capable of simulating performance in the actual task requires that all component processes be specified completely. The construction of UNDERSTAND, therefore, called for a substantial number of choices among alternative processing assumptions. In some cases, little or no empirical evidence was available for preferring one processing alternative to another.

Thus, several weakly motivated assumptions were, perforce, incorporated in the UNDERSTAND program. It is the main purpose of this paper to analyze evidence that bears on the validity of some of the more important of these assumptions. The principal evidence to be examined consists of verbal thinking-aloud protocols obtained in an experiment in which different subjects were presented with different, but isomorphic, forms of a problem. (We will define "isomorphic" presently.) Some additional supporting evidence was obtained from an experiment, also using these isomorphs of the same problem, in which protocols were not recorded, but subjects were allowed to use paper and pencil in working the problem. In this latter experiment, the notations adopted spontaneously by the subjects are our primary data.

The subjects' task was to understand written problem instructions in natural language. There is, of course, no unique way to describe a given problem in natural language. Moreover, for any particular problem, it is usually easy to construct problem isomorphs—that is, problems whose solutions and moves can be placed in one-to-one relation with the solutions and moves of the given problem. By comparing and contrasting protocol data from subgroups of subjects presented with different classes of isomorphs, we can begin to see what particular aspects of the instruction

This research has been supported by Public Health Service Grant MH-07722 from the National Institute of Mental Health and by National Science Foundation Grant GS-38533. The authors are grateful to Don Waterman and David Kieras for many discussions on the subject of this paper. Requests for reprints should be sent to Herbert A. Simon, Department of Psychology, Carnegie-Mellon University, Pittsburgh, Pennsylvania 15213.
text have major importance for task performance, and we can assess the validity of some of the basic assumptions incorporated in the UNDERSTAND program.

The problems used in this study were 13 variants of a single problem, all formally isomorphs of the Tower of Hanoi puzzle. (See Fig. 1 for an example). That is to say, successive situations in these task environments could be mapped, in one-to-one fashion, on arrangements of disks on pegs in the Tower of Hanoi puzzle; and legal moves for each of the problems could be mapped into legal moves of the Tower of Hanoi. In fact, any of the problems could have been solved by mapping it into the corresponding Tower of Hanoi problem and then solving the latter. No subject did this, and only two or three even thought of trying or noticed the analogy. The problems, then, were identical in formal structure, but differed in their “cover stories.”

PREDICTIONS OF THE UNDERSTAND PROGRAM

The UNDERSTAND program postulates that understanding written instructions for a problem involves constructing an internal representation (in long-term memory) of the problem space, and a set of operators for making "moves" from one situation to another in that problem space. Such an internal representation and such a set of operators constitute the principal inputs that a program like the General Problem Solver (Newell & Simon, 1972, chap. 8) requires in order to set to work solving the problem. In the present form of the theory, there is no feedback from solution attempts to understanding—the problem is assumed to be fully understood before the solution processes begin their work. This is one of the assumptions of the theory we wish to test. As an alternative, it is easy to imagine a feedback loop, so that, after initial interpretation of the problem instructions, unsuccessful attempts at solution reveal inadequate understanding of the instructions and lead to additional efforts of interpretation. Does understanding wholly precede solution attempts, as the present program assumes; or are these two processes intermingled, with the solution processes returning control, from time to time, to the understanding processes?

The UNDERSTAND program itself has two main components. The first is a set of language processes, which take the instruction text and, using both syntactic and semantic means, extract from it a deep structure. The second component is a set of construction processes, which take the deep structure produced by the language processes and construct from it the internal problem representation and the legal move operators. Again, language and construction processes are assumed to work sequentially, in that order, without feedback. We will have occasion to discuss here only some general features of these processes; a detailed description of the program has been published elsewhere (Hayes & Simon, 1974).
The language processes incorporate a more or less conventional case-grammar parser. The system does have capabilities, however, for tracing anaphoric references across sentence boundaries. When a particular portion of the text cannot be parsed, it is simply dropped from consideration. Apart from anaphoric reference, little contextual meaning is carried across sentence boundaries; for the most part, the text is processed "locally," sentence by sentence.

The construction processes combine information extracted from the text by the language processes with several kinds of information already stored in semantic memory. First of all, the representation-constructing processes have built into them the assumption that problem situations can be represented as list structures expressing the relations among the objects and attributes mentioned in the problem instructions, and specifically, mentioned in the sentences that describe the initial problem situation. Hence, under this assumption, the representation is not created from the entire problem text but from particular specific sentences in that text.

Second, the operator-constructing processes assume that the problem's legal move operators can be described in terms of a few basic types of operations on list structures, like "Transfer A from X to Y," "Exchange X with Y," "Copy A from X to Y," "Erase A at X," "Insert A at X," and so on. These are the standard types of operations that appear as the primitives in any list-processing language, and bear a strong resemblance, also, to the elementary types of "actions" postulated by Schank (1972) in his semantic parsing schemes. We shall look for evidences in our data for the subjects' use of these sorts of basic operations or actions, abstracted from the particularities of the specific task described in the instructions.

The operator-constructing processes extract these legal move definitions from the corresponding sentences of the task instructions, but interpret them in the light of the representation that has previously been generated by the representation-constructing processes. Hence, the final form these operators take is much more context-dependent than the form of the representation; the former depends on several portions of the problem text, the latter only on one or a few sentences. In the case of the "Monster problem" text of Fig. 1, the problem representation would be constructed by the UNDERSTAND program from information extracted from the third sentence of the instructions; while the operators would be constructed from the fourth sentence and the three conditions following it—but formulated in such a way as to operate correctly on the representation already constructed.

Finally, in handling anaphoric reference, the UNDERSTAND system must adopt conventions for naming objects. For example, in the initial situation described in Fig. 1, the "small monster" may also be referred to as the "monster holding the large globe." In the program as it is now
S1. Three five-handed extraterrestrial monsters were holding three crystal globes.
S2. Because of the quantum-mechanical peculiarities of their neighborhood, both monsters and globes come in exactly three sizes with no others permitted: small, medium, and large.
S3. The medium-sized monster was holding the small globe; the small monster was holding the large globe; and the large monster was holding the medium-sized globe.
S4. Since this situation offended their keenly developed sense of symmetry, they proceeded to transfer globes from one monster to another so that each monster would have a globe proportionate to his own size.
S5. Monster etiquette complicated the solution of the problem since it requires:
S6. (1) that only one globe may be transferred at a time,
S7. (2) that if a monster is holding two globes, only the larger of the two may be transferred, and
S8. (3) that a globe may not be transferred to a monster who is holding a larger globe.
S9. By what sequence of transfers could the monsters have solved this problem?

FIG. 1. Monster Problem 1.

constructed, the selection of a representation automatically determines which of these names will be applied (in fact, the former). This is an important assumption that we shall also wish to test from the data of our experiment. We shall discuss the psychological meaning of the assumption in a later section.

In the main series of protocols to be examined here, the problem isomorphs were created from a single basic "cover story"—all were Monster problems—by making carefully structured changes in the language of the sentences (see Fig. 2). Since these sentences are used in this problem by the UNDERSTAND program to construct the problem representation and define the operators, changes in their wording can be expected to alter the representation and operator definitions. Specifically, the UNDERSTAND program will represent certain of these isomorphs as Change problems (i.e., problems whose legal-move operator is of the form "Change X to Y"); while it will represent other isomorphs as Transfer problems (i.e., problems whose legal-move operator is of the form "Transfer A from X to Y").

As can be seen from Fig. 2, Sentence 3 of each problem refers to two classes of objects (e.g., in Problem 5, "monsters' names" and "globes"). Sentence 4 designates the objects of one class as fixed, of the other as variable (e.g., in Problem 5, the globe held by each monster is fixed, his name is variable). By Sentences 7 and 8, the legality of moves depends on the ordering of one of the attributes (e.g., in Problem 5, the names are ordered by length). If the ordering refers to the variable objects, the problem is a Transfer problem; if it refers to the fixed objects, the problem is a Change problem. Thus, Problem 5 is a Transfer problem.

A problem is constructed by selecting the two classes of objects and their relation (e.g., "holding"), assigning one class to be fixed and the other variable, and choosing the problem form (Transfer or Change). Once these choices have been made, the problem instructions are determined.
<table>
<thead>
<tr>
<th>Problem</th>
<th>Number</th>
<th>Type</th>
<th>Sentence 3</th>
<th>Sentence 4</th>
<th>Sentence 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>T</td>
<td></td>
<td>The small monster held the large globe.</td>
<td>... to teleport globes ... monster would have a globe proportionate to his own size.</td>
<td>If a monster is holding two globes ... can transmit only the larger.</td>
</tr>
<tr>
<td>2</td>
<td>T</td>
<td></td>
<td>The small monster stood on the large globe.</td>
<td>... to teleport themselves ... monster would have a globe proportionate to his own size.</td>
<td>If two monsters are standing on the same globe, only the larger ... can leave.</td>
</tr>
<tr>
<td>3</td>
<td>C</td>
<td></td>
<td>The small monster was holding the large globe.</td>
<td>... to shrink and expand the globes ... monster would have a globe proportionate to his own size.</td>
<td>If two globes are of the same size, only the globe held by the larger monster ... can be changed.</td>
</tr>
<tr>
<td>4</td>
<td>C</td>
<td></td>
<td>The small monster was holding the large globe.</td>
<td>... to shrink and expand themselves ... monster would have a globe proportionate to his own size.</td>
<td>If two monsters are of the same size, only the monster holding the larger globe can change.</td>
</tr>
<tr>
<td>5</td>
<td>T</td>
<td></td>
<td>The monster with the small name was holding the large globe.</td>
<td>... to transfer names ... monster would have a globe proportionate to the size of his name.</td>
<td>If a monster has two names, ... can transmit only the longer.</td>
</tr>
<tr>
<td>6</td>
<td>T</td>
<td></td>
<td>The monster with the small tail was holding the large globe.</td>
<td>... to transfer tails ... monster would have a globe proportionate to the size of his tail.</td>
<td>If a monster has two tails, ... can transfer only the longer.</td>
</tr>
<tr>
<td>7</td>
<td>C</td>
<td></td>
<td>The small monster was originally large.</td>
<td>... to shrink and expand themselves ... monster would have his original size back.</td>
<td>If two monsters are of the same size, ... only the monster who was originally larger can change.</td>
</tr>
<tr>
<td>8</td>
<td>T</td>
<td></td>
<td>The monster with the small name originally had the large name.</td>
<td>... to transfer names ... monster would have his original name back.</td>
<td>If a monster has two names, ... can transmit only the longer.</td>
</tr>
<tr>
<td>9</td>
<td>T</td>
<td></td>
<td>The small monster was originally large.</td>
<td>... to transfer sizes ... monster would have his original size back.</td>
<td>If a monster has two sizes, ... can transfer only the larger.</td>
</tr>
</tbody>
</table>

Fig. 2. The Monster Problems.
The reader can test his understanding of Fig. 2 by making up a full problem instruction for Problem 5, using the language in the figure to replace the language for Problem 1 in Fig. 1, and making the corresponding changes in the remaining sentences.

Readers familiar with the Tower of Hanoi problem can construct a set of instructions for that problem by taking pegs as the fixed objects and disks as the variable objects, and regarding the disks as ordered by size. An exactly isomorphic problem (but in Change rather than Transfer format) can be obtained by making pegs variable and disks fixed (i.e., assigning pegs to disks), but ordering again by disk size.

The authors have demonstrated in another series of experiments, concerned with problem difficulty, that Transfer problems can be solved by subjects nearly twice as fast as isomorphic Change problems (Hayes & Simon, in press). In those experiments the differences in solution times were traced to differences in the difficulty of executing comparison processes when applying the rules for a legal move. The authors also found that transfer of training, when problem isomorphs were presented successively, varied markedly with the pair of isomorphs presented. The authors proposed that these differences in transfer of training were determined largely by the process of formulating the rules for legal moves.

The UNDERSTAND program predicts that subjects will adopt different representations and operator definitions for isomorphic Transfer and Change problems; and these differences could explain the observed differences in problem difficulty and transfer of training. It is important, therefore, to see if direct evidence can be found to test the program's predictions. (Since the UNDERSTAND program, first constructed and debugged in the context of the Tea Ceremony problem of Fig. 3, has not yet been fully extended to handle the input language for all of the isomorphs of the Monster Problem, the predictions we make here from the theory involve a certain amount of hand simulation of the program's behavior.)

The assumptions embedded in the UNDERSTAND program that lead to these predictions, made before the experimental facts just described had been discovered, are by no means obvious. Recall that all the tasks under consideration are isomorphic, hence are capable of being represented in the same problem space and with the same operators for legal moves. Since some of these representations have now been shown to facilitate solving the problem, and others to delay solution, it might be supposed that an instruction understanding process would involve not only discovering some representation for a problem, but discovering a particular representation that was convenient or facilitative. Alternatively, there might be associated with each problem a canonical representation, independent of isomorphic transformations; and the UNDER-
1. In the inns of certain Himalayan villages is practiced a most civilized and refined tea ceremony.

2. The ceremony involves a host and exactly two guests, neither more nor less.

3. When his guests have arrived and have seated themselves at his table, the host performs five services for them.

4. These services are listed below in the order of the nobility which the Himalayans attribute to them:
   - stoking the fire,
   - fanning the flames,
   - passing the rice cakes,
   - pouring the tea,
   - reciting poetry.

5. During the ceremony, any of those present may ask another, "Honored Sir, may I perform this onerous task for you?".

6. However, a person may request of another only the least noble of the tasks which the other is performing.

7. Further, if a person is performing any tasks, then he may not request a task which is nobler than the noblest task he is already performing.

8. Custom requires that by the time the tea ceremony is over, all of the tasks will have been transferred from the host to the most senior of the guests.

9. How may this be accomplished?

**Fig. 3. A Tea Ceremony.**

STAND program could construct the identical canonical representation, whichever isomorph was presented.

The UNDERSTAND program predicts that neither of these will be the case; instead, the problem solver will adopt the first representation to which the surface description of the problem leads him, and this will be a good or poor representation depending on the characteristics of that surface description. This prediction derives, in turn, from two characteristics of the program: (1) the absence of feedback from the solution processes to the UNDERSTAND program, and (2) the "local" operation of that part of the UNDERSTAND program that constructs the representation from a limited portion of the text of the instructions. The prediction might be thought to conflict with findings (e.g., Bransford & Franks, 1971) that differences in surface structure are often lost when sentences are processed and stored in memory. However, there is no real contradiction here. The prediction does not assert that all surface information is retained, but simply that when isomorphic information structures are expressed in quite different natural language encodings, they may lead to correspondingly different deep structure interpretations.

Thus we will be chiefly interested, in this paper, in evidence relating to four aspects of the UNDERSTAND program: (1) the assumption that understanding will be achieved prior to the start of solution attempts,
(2) the assumption that subjects will represent legal moves in terms of basic, rather abstract, information processes that operate upon list structures, (3) the prediction that the verbal form of the instructions will affect the naming of objects in specified ways, and (4) the prediction that the representations used in Change and Transfer problems will differ in specified ways. With these explanations of the purposes of this study, and the nature of the assumptions embedded in the UNDERSTAND program and their consequences, we turn now to the data and their analysis.

MATERIALS AND METHODS

The Data

Each of 20 subjects was given a one-page description of a puzzlelike problem, and asked to solve it, thinking aloud as he worked. The recorded verbal protocols for these 20 problems constitute the data that we shall analyse.

The subjects were male and female graduate students and faculty members at Carnegie-Mellon University. They were not previously familiar with the particular problems on which they worked, although a few of them had encountered at least one problem that was formally isomorphic to the one on which the protocol was taken. In only one case could a subject be described as "experienced" with problems of this kind. In any case, there was no evidence that their limited experience affected their behavior in the task, and some positive evidence that it did not—at least along dimensions relevant to our present discussion.

The tasks were the 13 problem isomorphs described above, providing two or three different protocols for a few problem variants, and a single protocol for each of the remaining variants. Since the problems are all formal isomorphs, we can regard them, for purposes of analysis, as a sample of tasks drawn from a well defined population. Lacking a taxonomy of isomorphs in this populations, there is no way to sample from it randomly. Instead, we have simply generated a collection of isomorphs that vary along a number of dimensions and have carried out our analysis (1) by examining properties that are shared by essentially all the isomorphs, and (2) by dividing the isomorphs into dichotomous subclasses according to variables we had reason to think were important and comparing the properties of these subclasses. In this way we are able to establish some of the differences between the subclasses as being statistically significant.

Figure 1 gives the problem text for the variant called "A Monster Problem 1." Figure 2 describes variants of the Monster problem that we will have occasion to examine in detail. Figure 3 gives the problem text for the variant called "A Tea Ceremony," while Fig. 4 is a complete list of problems and subjects.

Method of Analysis

The methods that we have used previously for analysis of problem solving protocols—in particular, the problem behavior graph—are not suitable for the present undertaking. Use of the problem behavior graph to depict and summarize a subject's behavior requires that the subject operate in a well-defined problem space. This requirement is generally only met when subjects are confronted with well-structured problems of a sort with which they are already familiar (Newell & Simon, 1972).

In the present instance, however, it is precisely the process whereby the subject first constructs a new problem space in the face of a novel problem that chiefly interests us. In this experiment, the subjects had no prior experience with the problem instructions.
Understanding those instructions—constructing a problem space, describing the initial problem situation, and constructing operators for changing the situation (making moves)—constituted their first task. We wish to find in the protocols evidence that shows what kinds of problem descriptions the subjects built and how they went about building them.

The principal evidence we have found in the protocols for identifying the subjects’ problem spaces are: (1) the frequency with which they reread each particular sentence in the problem statement prior to their first move attempt; (2) their verbal exchanges with the experimenter, initiated either by subject or by the experimenter; (3) the way in which subjects described moves and named objects in their move attempts; and (4) the way in which subjects described the current situation before or after a move attempt. In addition, the protocols were examined for explicit comments of the subjects that might reveal something about their representations of the problems.

At the end of the solution attempt, whether successful or not, subjects were debriefed by the experimenter, and we have found some of the comments during debriefing to be illuminating. In general, however, we have relied upon the subjects’ thinking-aloud behavior while they were actually working the problems as the source of our evidence, rather than upon their subsequent introspections of retrospections about what they had been doing.

RESULTS AND DISCUSSION

Reading the Instructions

Among the 20 protocols, there were 28 occasions when a subject reread the sentence describing the initial problem situation (Sentence 3) prior to his first attempted move. (See Table 1.) There were 42 occasions when a subject reread one or more of the sentences in the problem instructions that defined the conditions for legal moves (Sentences 6 through 8). There were 10 occasions when a subject reread the sentence (Sentence 4) de-
TABLE 1

NUMBER OF TIMES INSTRUCTIONS WERE REREAD PRIOR TO SOLUTION ATTEMPTS

<table>
<thead>
<tr>
<th>Problem</th>
<th>Number of protocols</th>
<th>Number of rereadings of sentence</th>
<th>S3</th>
<th>S4</th>
<th>S6–8</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monster problems</td>
<td>14</td>
<td>23</td>
<td>9</td>
<td>32</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Other problems</td>
<td>6</td>
<td>5</td>
<td>1</td>
<td>10</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>All problems</td>
<td>20</td>
<td>28</td>
<td>10</td>
<td>42</td>
<td>6</td>
<td></td>
</tr>
</tbody>
</table>

fining the problem goal, there were six occasions when a subject reread one of the other sentences (Sentences 1, 2, or 9).

A subject will presumably reread a sentence if (1) he believes the information in the sentence is relevant to his task, and (2) he does not, at present, retain that information in short-term or long-term memory. For example, in the Monster Problems, we observe that Sentences 1 and 2 contain mostly irrelevant information; all that needs to be retained from these sentences is that three monsters are holding three globes, and that monsters and globes each come only in small, medium, and large sizes. The rest of the information is irrelevant, and even the relevant information is largely redundant, being repeated in later problem statements. Sentence 9, similarly, contains only an easily remembered statement of the task. These three sentences are reread only infrequently. In subsequent work to be reported elsewhere (Hayes & Waterman, Note 1), a simulation of the relevance-judging process, called ATTEND, made essentially the same judgments about Sentences 1, 2 and 9.

Sentence 4, while relevant (at least in its second clause), describes the goal in terms of a single simple condition (e.g., that globe sizes should correspond with monster sizes), which is easily remembered. This sentence, also, is reread only infrequently.

Sentence 3, on the other hand, is relevant and, moreover, involves three pairings of objects which are listed (except in five of the variants of the Monster Problem) in an arbitrary order not corresponding to the order of sizes of either set of objects. The conditions of Sentences 6 through 8 (and especially 7 and 8) were similarly complex, and correspondingly difficult to retain in memory. Subjects had to return to all of these more complex sentences. We conclude that frequency of rereading depends mainly on the relevance of sentences and the difficulty of holding their contents in memory.

After the subjects have begun to make moves, they still need sometimes to refer back to the problem instructions. These references are predominately to the move conditions, Sentences 6 through 8, but occasionally also to the initial situation, Sentence 3. Virtually no other
sentences are reread after the subjects have actually begun to make trial moves in efforts to solve the problems.

The only aspect of these particular findings that is nonobvious is that subjects appear to be able to make judgments about the relevance of particular sentences in the problem statement at a relatively early stage (see also Hayes & Waterman, Note 1, for additional evidence of this ability). The UNDERSTAND program behaves in a similar fashion, for it only continues to process those sentences (3 through 8) that are relevant to the construction of the representation and operators. The program (and perhaps also the human subjects) can do this because its programs and stored semantic information tell it what it is looking for, and it need not depend upon explicit indications in the problem instructions to separate wheat from chaff.

**Exchanges between Subject and Experimenter**

If, during the taking of a protocol, the subject asked a question, the experimenter replied briefly. If the subject made an illegal move that was noticed by the experimenter, and the subject was proceeding to carry out further moves without checking, the experimenter interrupted to ask him to recheck the legality of the last move. A total of 76 exchanges between experimenter and subject were noted in the 20 protocols. If we classify these exchanges by their content, rather than by who initiated them, we find that 40 were Task exchanges, resulting from questions or confusions the subject had about the exact nature of the task. The remaining 36 exchanges may be classified as Application exchanges because they occurred after misapplications of the rules that produced illegal moves. All of the Task exchanges and some of the Application exchanges were initiated by the subjects. (See Table 2)

In 12 protocols, representing 71 of the total of 76 exchanges, examples of both Task and Application exchanges occurred. In only seven cases among these 71 exchanges did a Task exchange occur after the first Application exchange. In four of the seven cases, when this happened the subject started a new solution attempt from the beginning.

**TABLE 2**

<table>
<thead>
<tr>
<th>Type of exchange</th>
<th>Number of protocols</th>
<th>Number of exchanges</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Kinds</td>
<td>20⁹</td>
<td>76</td>
</tr>
<tr>
<td>Task</td>
<td>14</td>
<td>40</td>
</tr>
<tr>
<td>Application</td>
<td>13</td>
<td>36</td>
</tr>
<tr>
<td>Both</td>
<td>12</td>
<td>71</td>
</tr>
</tbody>
</table>

* Includes five protocols having no exchanges.
These data lend themselves to a rather straightforward interpretation. In general, subjects did not attempt a solution of the problem until they thought they understood the instructions. As postulated by the UNDERSTAND program, the understanding processes usually preceded the solving processes. In achieving their understanding, most subjects (15 of the 20) asked one or more clarifying questions of the experimenter.

Contrary to the assumption of the UNDERSTAND program, however, there was some evidence of feedback from the solution-seeking processes to the understanding processes, and a consequent interweaving of the two. Many subjects, when they attempted to undertake a solution of the problem, made illegal moves, either because they had forgotten the conditions or the current situation, or (but less frequently) because they did not, in fact, have an accurate understanding of the problem instructions. In the latter case, recognition of the error often led to reinterpretation of the task instructions (and sometimes new Task questions), whereupon the subject began his solution attempts over again. As noted above, however, there were only seven instances of this interweaving of the two processes.

Of the 40 Task questions, three-quarters (31) refer to just seven topics (Table 3). Three subjects asked if the problem had a unique solution. Three asked whether they might use pencil and paper. Although Sentence 6 gives a clear negative answer, five asked whether two or more moves could be made simultaneously (e.g., whether globes could be "exchanged" between monsters). Five asked whether a move from "large" to "small" had to pass, by two steps, through "medium." Five asked questions about the meaning of the ordering relation in Sentences 7 and 8. Seven asked whether two or more globes could be held simultaneously (i.e., whether there was always a one-one relation between monsters and globes). The three subjects doing the Tea Cere-

<table>
<thead>
<tr>
<th>Question</th>
<th>Number of Questions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Can hold several globes simultaneously?</td>
<td>7</td>
</tr>
<tr>
<td>Simultaneous moves (exchanges) legal?</td>
<td>5</td>
</tr>
<tr>
<td>Must moves pass through &quot;medium&quot; (continuity)?</td>
<td>5</td>
</tr>
<tr>
<td>Meaning of ordering relation?</td>
<td>5</td>
</tr>
<tr>
<td>Unique solution?</td>
<td>3</td>
</tr>
<tr>
<td>Paper and pencil?</td>
<td>3</td>
</tr>
<tr>
<td>Meaning of &quot;'senior guest'&quot;?</td>
<td>3</td>
</tr>
<tr>
<td>All other</td>
<td>9</td>
</tr>
<tr>
<td>Total</td>
<td>40</td>
</tr>
</tbody>
</table>

TABLE 3
Number of Task Questions, by Topic
mony problem all asked the meaning of "senior guest." The remaining
nine questions covered a wide range of topics without any particular
discernible pattern.

All of these questions can be interpreted as responses to ambiguities in
the task instructions (only occasionally do subjects fail to notice infor-
mation that is explicit). Except for the question about the term "senior
guest," the questions were not specialized to these particular problems;
rather they were mostly questions about how the problem moves should be
interpreted semantically, but at a relatively abstract level. Was simulta-
neity excluded? Was continuity implied? Was a relation one-one? Was a relation exclusive?

The abstract character of these questions is consistent with the
assumption of the UNDERSTAND program that the subject is trying to
map the concrete description of the problem situation onto an
abstract list-structure representation, and the description of legal moves
onto the basic actions stored in semantic memory. The five questions
asked most frequently fit this interpretation. (See Table 3). One is a
question as to whether the relation between the two types of objects
(globes and monsters) is one-one or one-many. A second asks whether
the basic legal move is to be classified as an exchange (simultaneous
moves) or a transfer. A third asks whether the ordering on sizes implies
that sizes must change continuously (i.e., pass through "medium" in
going from "large" to "small"). A fourth seeks to clarify the
restrictions placed on legal moves by the ordering relation. A fifth asks
whether there is a single path to the goal in the problem space, or
several paths.

The Problem Representations

The central objective of our inquiry is to determine how subjects
represented the problem space that they generated in order to try to solve
their problems. In particular, did changes in the problem statement, while
retaining isomorphism, change the processing of the text in such a way
as to change the representation that the subject created and used in his
solution attempts. In answering this question, we will limit ourselves to
the 14 protocols for isomorphs of the Monster problems. Because the
cover stories for these isomorphs are highly similar, it is easy to charac-
terize the main differences among them (see Fig. 2 and Table 4) and to
sort them in dichotomous classes on the basis of salient characteristics,
whose effect upon the representations chosen can then be tested.

Consider a Monster problem in which the problem instructions describe
monsters as holding globes. There are at least two obvious ways in
which this problem can be represented by means of list structures: (1) a list
of Monsters is kept in memory, and with each Monster, a list of the
globes he is holding; or (2) a list of globes, and with each globe the
<table>
<thead>
<tr>
<th>Protocol</th>
<th>Primary object</th>
<th>Ordering of S3</th>
<th>Predicted from program</th>
<th>Actual moves</th>
<th>Actual situations</th>
<th>Predicted Initial</th>
<th>Actual Initial</th>
<th>Subsequent Initial</th>
</tr>
</thead>
</table>

Notes. M1 through M9 denote Monster Problems 1 through 9, respectively. Perm. — Permanent; Var. — Variable.
a Ordered, left to right, by size of globes.
b Referred to monsters as occupying "slots" in order of their current size, and as moving from slot to slot.
c Numbered monsters in order of mention in S3.
d Represented size by vertical displacement.
e Explicitly and unsuccessfully tried to use visual imagery.
f Kept track by position on instruction sheet.
g Used finger to keep track of place on instruction.
h Vacillated between variable and permanent naming.

name of the monster (as the value of an appropriate attribute) who is holding it. More briefly: globes can be associated with monsters, or monsters with globes.

Which of these two representations is chosen will determine, in turn, how moves are made. Consider the process of moving Globe A from Monster 1 to Monster 2. With the first representation, Globe A must be dissociated from Monster 1 and associated with Monster 2. With the second representation, the value of the monster associated with Globe A must be changed from 1 to 2. We will call the former representation a Transfer representation, and the latter a Change representation. With a Transfer representation something (e.g., a globe) moves from one location (a monster) to another. With a change representation, the object (monster) associated with a particular location (globe) is altered.
**Names for objects.** Closely related to the form of the representation is the way in which the objects that appear in the problem can be named. In Monster Problem 1, for example, we can speak of "the large monster" or "the monster holding the small globe." The former name is **Permanent,** for in this problem, the size of a monster does not change as moves are made. The latter name is **Variable:** it changes as the globes are passed from one monster to another, for then different globes are associated successively with the same monster. Thus, for example, the monster who was initially holding the small globe may later become the "monster holding the large globe."

The name that is given to an object by Sentence 3, in the description of the initial problem situation, will be called its **Initial** name. Regardless of whether the object is named permanently or transiently, the problem solver has the option of referring to it by its Initial name, and the protocols show that that option is often used. Thus we find many statements of the form: "The globe that was initially small." In sum, at any given time while he is working on a problem, a subject may refer to an object by its Permanent name, its Initial name, or its Variable name; or by some combination of these (e.g., "the large monster that is now holding the medium-size globe," or "the monster who was at the outset holding the large globe and is now holding the small globe.")

Sentence 3 in each of the Monster problems paired one set of things (e.g., monsters) of different sizes with another set of things (e.g., globes) of different sizes. In most of the problem statements, the sizes of both sets of things were listed in an irregular order—c.g., large; small; medium, but not small; medium, large. In five of the fourteen sets of instructions, however, the sizes of the things mentioned in the predicates were ordered from small through medium to large. (See Table 4, Column 3.) As we shall see, subjects found it easier to keep track of the successive problem situations in the ordered than in the unordered condition, and adopted some special naming tactics when they were confronted with the latter condition.

In the problem instructions, objects may be referred to by either Permanent or Variable names. For example, Sentence 3 of Monster Problem 1 contains the clause, "the small monster held the large globe." In this variant of the problem, neither monsters nor globes change size, so the naming of both is permanent. In Monster Problem 4, however, monsters change their sizes; hence the identical clause in Sentence 3 refers to the monster by a Variable name. If the subject wishes to employ a Permanent name for a monster in this problem variant, he must use a phrase like "the monster holding the large globe," for monsters keep the same globe throughout the problem. Table 4, column 7, shows, for Sentence 3 of each variant of the Monster problem, whether the subject of that sentence assigns a Permanent or a Variable
name to its referent. In all cases where the sentence assigns a Variable name, the referent can be named permanently by referring to it in terms of the corresponding predicate—as in the example for Monster Problem 4, above. It will be observed that the subjects of the clauses of Sentence 3 in Problems 1, 2, 3, and 8 provide Permanent names for their referents, while the corresponding clauses in the other problem statements provide Variable names. Four subjects solved problems in which the Initial name was Permanent; 10 solved problems in which it was Variable.

Transfer problems and change problems. The variants of the Monster problems can also be dichotomized in another way. Sentence 4 of Problem 1, which defines what is meant by a "move," states: "they proceeded to teleport globes from one to another." The UNDERSTAND program formalizes the move as a three-argument relation: $\text{TRANSFER}(G, M_1, M_2)$, where the first argument denotes a globe, and the other two arguments monsters. In Monster Problem 3, the corresponding clause of Sentence 4 is: "they proceeded to shrink and expand the globes," which the UNDERSTAND program formalizes as: $\text{CHANGE}(G, S_1, S_2)$, where the first argument denotes a globe, and the other two arguments sizes.

In spite of the formal similarity between the moves of the two problems—both represented by three-argument relations—there is an important difference between them. The rules governing "moves" (Sentences 4 through 8) in Monster Problem 3 make no reference to the monsters who are holding the globes. Suppose, now, that the subject represents the situation in either of these problem variants as a list of monsters, and, associated with each member of the list, a list of the globes he is currently holding. Each monster would be further labeled by his size, as would each globe. Thus, the representation might look like this:

Monster-list
Monster 1
Size: Medium
Globes
Globe A
Size: Large

Monster 2
Size: Large
Globes
Globe B
Size: Small

Monster 3
Size: Small
Globes
Globe C
Size: Medium.
Now, to perform the operation \textsc{transfer}(G_A, M_1, M_2), we must locate \(M_1\) and \(M_2\) on the Monster-list, remove Globe A from the list of globes of Monster 1, and add Globe A to the list of globes of Monster 2. The operation is one of \textit{transferring} a symbol from one list to another. On the other hand, to perform the operation \textsc{change}(G_A, \text{Large, Small}), we must search the list of monsters until we find the one holding Globe A, then change the value of the size attribute of that globe from large to small. The operation is one of \textit{changing} the value of an attribute of an object.

The \textsc{understand} program, reading the problem instructions of a Monster problem, would construct a representation from the information in Sentence 3, and the particular representation chosen would depend upon the grammatical structure of that sentence. If the clauses of that sentence describe monsters holding globes, then the situation would be described as a list of monsters, and associated with each monster, a list of the globes it was holding. In Problems 1 and 3, for example, the program would create precisely the representation that is depicted above. Because the form of the moves in these two problems is determined, as we have seen previously, by the structure of Sentence 4 (and the similar structures of Sentences 6 through 8), the \textsc{understand} program would interpret Problem 1 as a \textsc{transfer} problem, and Problem 3 as a \textsc{change} problem.\footnote{The assertions of this paragraph and the next are based upon a careful hand simulation of the \textsc{understand} program with these problem texts. We are now undertaking to verify these conclusions automatically by expanding the vocabulary and syntax of \textsc{understand} so that it will handle all of the problem variants.}

In Table 4, column 4, the variants of the Monster problems are classified as \textsc{transfer} problems or \textsc{change} problems, respectively, in the manner we have just described. First, a representation is induced directly from Sentence 3; then the legal move described in Sentence 4 is described in terms of this representation. With this method of classification, Problems 1, 2, 5, 6, 8, and 9 become \textsc{transfer} problems, while Problems 3, 4, and 7 become \textsc{change} problems. If the \textsc{understand} program represents correctly the way in which human subjects go about interpreting the problem instructions, then we must predict that the same classification of the problems will be inferable from the data of the subjects' protocols. Eight subjects solved \textsc{transfer} problems, six solved \textsc{change} problems.

\textit{Legality tests}. The distinction between a \textsc{transfer} problem and a \textsc{change} problem has a further consequence. Not only are different processes required to make a move in the two kinds of problems, but the tests for legality of a move, defined by Sentences 6 through 8 of the problem instructions, are different also. In the representation above, the legality of the move, \textsc{t}(G, M_1, M_2) depends on the size of G compared with the sizes of other globes assigned to \(M_1\) and \(M_2\). To perform these tests, the
globe lists of M1 and M2 must be accessed. On the other hand, the legality of the move, C(G, S1, S2), is ascertained by finding all the globes that are of size S1, and determining if G is held by the largest of the corresponding monsters, then finding all the globes that are of size S2, and making the same determination. Thus, it is a more complex and lengthier processing task to determine the legality of a move with the Change representation than with the Transfer representation, although it is perhaps no more complex actually to make the move once its legality has been determined. Figure 5 shows more precisely just why testing the legality of a Change is more complex than testing the legality of a Transfer.

In the Transfer operation, we test, for all other globes belonging to Monster 1, whether Globe A is greater than or equal to the other globe. If the test fails, the transfer is illegal (Rule 2). If the test succeeds in all cases, we test, for all globes belonging to Monster 2, whether Globe A is greater than or equal to the other globe. If the test fails, the transfer is illegal (Rule 3). If the test succeeds in all cases, the transfer is legal, Globe A is deleted from the globes of Monster 1, and added to the globes of Monster 2.

In the Change operation, we test, for all globes other than Globe A, whether these globes are equal in size to S1. For each case where this test succeeds, we test whether the monster of Globe A is at least equal in size to the monster of the other globe. If the latter test fails, the change is illegal (Rule 2). If the tests all succeed, we test, for all globes other

TRANSFER(Globe, A, Monster, 1, Monster, 2)
FOR-ALL-OTHER(Globe, A, Globe, Monster, 1):
    TEST-GE(SIZE(Globe, A), SIZE(Globe, Monster, 1));
    IF TEST FAILS: RETURN "ILLEGAL"; HALT;
FOR-ALL(Globe(Monster, 2)):
    TEST-GE(SIZE(Globe, A), SIZE(Globe(Monster, 2)));
    IF TEST FAILS: RETURN "ILLEGAL"; HALT;
DELETE(Globe, A, Globe(Monster, 1));
ADD(Globe, A, Globe(Monster, 2)); HALT.

CHANGE(Globe, A, S1, S2)
FOR-ALL-OTHER(Globe, A, Globe):
    TEST-ED(SIZE(Globe), SIZE, 1);
    IF TEST SUCCEEDS: CALL GLOBE GLOBE, B;
    TEST-GE(SIZE(Monster(Globe, A), SIZE(Monster, Globe, B)));
    IF TEST FAILS: RETURN "ILLEGAL"; HALT;
    IF TEST FAILS: GO NEXT;
FOR-ALL-OTHER(Globe, A, Globe):
    TEST-ED(SIZE(Globe), SIZE, 2);
    IF TEST SUCCEEDS: CALL GLOBE GLOBE, B;
    TEST-GE(SIZE(Monster(Globe, A), SIZE(Monster, Globe, B)));
    IF TEST FAILS: RETURN "ILLEGAL"; HALT;
    IF TEST FAILS: GO NEXT;
SET-ED(SIZE(Globe, A), SIZE, 2); HALT.

FIG. 5. Programs for Transfer and Change operators. "Test-Eq (X, Y)" means "Test if X is equal to Y." "Test-GE (X, Y)" means "Test if X is greater than or equal to Y." "For-all-other (X, Y)" means "for all the Ys other than X." "X (Y)" means "the X(s) of Y."
than Globe A, whether these globes are equal in size to S2. For each case where this test succeeds, we test whether the monster of Globe A is at least equal in size to the monster of the other globe. If the latter test fails, the change is illegal (Rule 3). If the test succeeds in all cases, we set the size of Globe 1 equal to S2.

In the Change operator, we have sets of nested tests, while in the Transfer operator, we do not. The UNDERSTAND program predicts that the form of Sentence 3, in relation to form of Sentence 4, will determine whether a subject will represent a problem as a Transfer problem or a Change problem, and hence, that the forms of these two sentences will be a major determinant of problem difficulty.²

There is nothing that requires a subject to represent any problem in the particular way we have indicated. In Monster Problem 3, for example, the situation could be represented as a list of globe sizes, and associated with each member of that list the globes of that size (the globes would be named, permanently, by the names of the monsters holding them). In this representation, the problem becomes a Transfer problem instead of a Change problem.

It would be possible for a subject to seek that representation which is simplest, according to some criterion, or to translate all such problems into the same, canonical, representation. The postulate incorporated in the UNDERSTAND program, however, is that subjects will not employ such alternative strategies, even though they are available, but will adopt the representation that constitutes the most straightforward translation of Sentences 3 and 4 of the instructions.

Empirical Evidence for Representations and Naming

In this section, we will compare the predictions with respect to problem representation and naming made by our hand simulation of the UNDERSTAND program with the behaviors of the human subjects as inferred from their protocols. We will first take up the evidence for representations, and then for naming.

The evidence was gathered by coding each instance in the protocols where the subject described a situation, mentioned a move, or named one of the objects in the problem. The entire protocol was then classified, according to the representation the subject used and his naming practices, on the basis of the codings of the majority of the relevant instances. In almost all cases, the great preponderance of instances fell in the one category or the other; close divisions were rare. In three protocols where there was variability in the subjects' naming practices,

² In studies to be reported elsewhere, the authors have found that Transfer problems require an average of 16.9 min for solution while Change problems require an average of 28.1 min.
we classified the protocols in the way that would be least favorable to the hypotheses we were testing. Hence, the statistics and significance levels given below are conservative.

Initially, the protocol elements were coded by a single judge, and later, independently, by a second judge. The two judges agreed on more than 90% of the individual protocol elements. With only four exceptions, they agreed fully on the classifications of the protocols. Three of the exceptions were the instances, mentioned above, where the subject was not consistent in naming objects by their Variable or Permanent names. The judges did not disagree on these protocols about the codings of individual elements, but found it difficult to classify the protocols as wholes. We have indicated above how we handled this problem. Finally, in one instance the second judge did not find enough evidence of the way in which situations were represented by the subject to classify the protocol from this evidence. However, on the basis of the subjects' descriptions of moves, both judges determined that this subject was using a Change representation of the problem.

**Representations.** We must ask first how we can determine empirically what representation a subject is using. If we limit ourselves to the initial portions of the protocols, the portions prior to the first attempted move, we find relatively little concrete evidence about the nature of the representation the subject is constructing. This portion of the protocol typically consists largely of reading and rereading of the problem instructions, together with questions addressed to the experimenter about their meaning.

Once the subject begins to attempt moves, however, he provides much richer evidence about the representation he has constructed. This evidence comes from the way in which he names the moves, together with the way in which he describes the situations that are created by the moves. Consider the statement, "So the small globe goes over to the guy with the medium-size globe." This sentence describes globes as moving from one monster to another, hence implies a list of monsters, each with associated globes. Consider next the statement from a different protocol, "The medium-size monster changes his into a small globe." Here again, globes are the possessions of monsters; they change in size without changing ownership. Finally, consider, "I will move the monster which is on the medium-sized globe to the large globe." Here the relation is reversed: There is a list of globes and monsters are associated with globes, instead of vice versa.

The evidence from the way in which situations are described is equally clear. In a protocol from Problem 1: "The small monster has the large, the medium has the small, and the large has the medium." A protocol from Problem 4: "A small monster is holding a small globe, a large monster holding a medium-size globe, and a medium-sized monster
holding a large globe." Or consider Monster Problem 5, in which the monsters transfer names, and a monster can have more than one name (or none) at a time: "The guy with the long name now has a long name and a medium-size name."

Table 4, columns 5 and 6, summarizes the evidence about the subjects' representations from both their move statements and their situation descriptions. The Table shows that both sources of evidence imply the same representations in all cases. It shows further that, except for Monster Problem 2 (monsters standing on globes), monsters are always selected as the primary objects—i.e., the main list in the representation—then are further described by the globes they hold, their sizes, their names, and so on.

There is another and more fundamental way of looking at the representations. We have seen that, by virtue of Sentence 4 of the problem instructions, the UNDERSTAND program describes Problems 1, 2, 5, 6, 8, and 9 as Transfer problems, and Problems 3, 4, and 7 as Change problems. In all cases except one (Subject L. A. on Problem 7), that is also the way in which the subjects represented them internally. The exceptional case, Problem 7, stated in the instructions as a Change problem, was transformed by Subject L. A. into a Transfer problem. There were no transformations in the other direction. The sample of one is too small to draw firm conclusions, but it is suggestive that the one example of problem transformation that we found was in the direction from the more difficult to the easier form of the problem.

It was observed earlier, and can be seen from the data in Table 3, that most of the Task questions subjects asked were of a rather abstract nature, referring to formal properties of the relations among objects in the problem rather than to their concrete semantic interpretation. This, of course, was an appropriate reaction to these problems, whose solutions do not depend on the laws of quantum mechanics, the sizes of globes, or the social customs of monsters. The appropriate strategy was to strip away such semantic meanings, and to abstract the problems down to their bare essentials. As several subjects pointed out explicitly, terms like "quantum-mechanical" and the fact that monsters were described as five-handed were simply irrelevant.

Some additional studies to be reported in detail elsewhere (Hayes & Waterman, Note 1) indicate that subjects can discriminate between the "real" problem and the "cover story" quite early in the course of their exposure to the problem text. Subjects, viewing the text sentences one at a time, judged the relevance of each for solution of the problem. For

---

3 The departure from the null hypothesis of equally likely agreement or disagreement is significant at the .00006 level.
4 Again, this agreement is significant at the .00006 level.
5 Agreement with the theory in 13 cases out of 14 is significant at the .00009 level.
the Tea Ceremony, subjects’ judgements of relevance were correct in 77% of the cases on a first reading and 86% on second reading. Corresponding percentages for the Monster problem were 77 and 90%.

If the problems we had presented the subjects had been genuine problems in physics, or some other real-world subject, the abstracting behavior might not have been appropriate. In that case, it would have been necessary to retain the physical content of the problems, and to draw upon knowledge of physics stored in long-term memory. What cued the subjects to respond in an abstract manner? We do not know for sure. It may have been the fact that these problems resembled other puzzlelike problems they had known, and which required an abstracting strategy. It may have been the fact that the problems were presented in a psychological laboratory and not in a physics class. It may have been that the problem instructions appeared, on their face, to describe puzzles rather than genuine real-world problems.

Whatever the reason, 12 of the 14 subjects who were given Monster problems adopted a strategy of abstraction from the beginning. Two subjects, however, did not. One regarded the phrase “quantum mechanical” as probably significant, and introduced notions of energy conservation in his attempts to solve the problem. A second subject avoided the complications of Sentences 6 and 8 by allowing his monsters to enter into a mutual pact to define new sizes to which they could change. Then no two monsters would become the same size. We have no very convincing evidence that would tell us why these particular subjects behaved differently from the others. Both of the exceptional subjects are trained in the humanities, and probably do not encounter many formal, abstract problems in their professional activities. However, a number of the other subjects who immediately adopted abstract interpretations of the problems were also trained in the humanities. We simply mention these individual differences to indicate that there is nothing inevitable about the application of an abstracting strategy, and that a fuller study of individual differences in this dimension of problem-solving style might be instructive.

Further evidence on representations. In two experiments with Monster problems designed to test relative difficulty of Transfer and Change problems, and the amounts of transfer of training from one to the other, subjects did not provide thinking-aloud protocols, but were allowed to use paper and pencil to record their successive moves (Hayes & Simon, in press). In other respects, these experiments were conducted in the same way as those reported in detail in this paper.

In these experiments, more than half of the subjects spontaneously and without instruction or training adopted a notation ("state-matrix" notation) that permits an unequivocal determination of whether they were using a Transfer representation or a Change representation of the problem. Of 62 subjects who were presented with a Transfer version of the problem,
The understanding process

<table>
<thead>
<tr>
<th>Transfer Type</th>
<th>Change Type</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>M</strong></td>
<td><strong>L</strong></td>
</tr>
<tr>
<td>0</td>
<td>L</td>
</tr>
<tr>
<td>1</td>
<td>—</td>
</tr>
<tr>
<td>2</td>
<td>M</td>
</tr>
</tbody>
</table>

Fig. 6. State-matrix notations used by subjects. (6A) shows the notation used in solving Transfer problems, (6B) the notation used in solving Change problems. The columns correspond to the fixed attribute; the rows to the successive problem situations after each move (0 is the starting situation). Within the cells are shown the current values of the variable attributes, which either (a) migrate from column to column (Transfer type) or (b) change value within a column (Change type).

37 used the state-matrix notation for recording their moves; of 55 subjects who were presented with a Change version of the problem, 30 used the state-matrix notation.

With the state-matrix notation, the problem solution is described by specifying the state of all the problem elements at each step of the solution. Most frequently the subjects portrayed the sequence of states in a matrix like one of those shown in Fig. 6. Each column of the matrix represents an object of one type (globe or monster); entries in the matrix represent the object or objects of the other type associated with a column. Rows represent successive problem states after each move. Thus, in Fig. 6A, at the starting state (State 0), the large object of Type 2 is associated with the medium object of Type 1; while at State 1, the large and small objects of Type 2 are associated with the large object of Type 1 (representing a transfer of the large Type 2 object from the medium to the large Type 1 object).

A matrix like that in Fig. 6A gives clear evidence that the subject who employs it has a Transfer representation of the problem; while a matrix like that in Fig. 6B just as clearly denotes a Change representation. Note that in Fig. 6A, relations between objects of Type 1 and Type 2 may be one-many, while in Fig. 6B, they are many-one.

Now the striking finding of these experiments is that, without a single exception among the 67 subjects who used the state-matrix notation, every single one used the form of the notation consistent with the form of the problem presented him. Hence all 37 subjects who were given Transfer forms of the Monster problem gave evidence of adopting a Transfer representation of the problem; while all 30 who were given Change forms of the problem gave evidence of adopting a Change representation.

**Naming.** Table 4, some of whose contents are summarized in Table 5,
also tells us something about the way in which the subjects name objects. With three exceptions in 14 protocols, in the early portions of the protocols, the name is determined by the form of Sentence 3. In those cases where the name so determined is a Variable name, the method of naming is usually (5 cases in 8) changed by the subject in the course of his problem-solving efforts so that the objects are subsequently referred to by Permanent names. There are no examples in the data of a progression in the opposite direction.

Thus, we see in Table 5 that, in eight of the 10 cases where the grammatical subject of Sentence 3 was a Variable name, the initial name applied to it by the subject was also Variable. In five of these eight cases, however, the subject at some later point in the protocol was found referring to the same object by a Permanent name. In only one of the four cases where the grammatical subject of Sentence 3 was a Permanent name was it referred to initially by a Variable name, and in all four cases it was referred to later by a Permanent name. There were no differences in initial naming practice between Transform problems and

<table>
<thead>
<tr>
<th>Type of Problem</th>
<th>Initial name</th>
<th>Intermediate or final name</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Var.</td>
<td>Perm.</td>
</tr>
<tr>
<td>Transfer-Permanent</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Transfer-Variable</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Change-Permanent</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Change-Variable</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>All Variable</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>All Permanent</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>All Transfer</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>All Change</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>All Problems</td>
<td>9</td>
<td>5</td>
</tr>
</tbody>
</table>

*Note.* Var. — Variable name; Perm. — Permanent name.

* For the distinction between Transfer and Change problems, see text. "Variable" problems are those in which the subject of Sentence 3 has a Variable name; "Permanent" problems, those in which the subject of Sentence 3 has a Permanent name.

* The hypothesis that the naming is independent of Sentence 3 is disconfirmed (11 cases out of 14) at the .028 level.

* Again, if Permanent and Transient naming are equally likely, the chance of using a Transient name three or fewer times in 14 is only .028.
Change problems—the important variable was the form of Sentence 3. In sum, the initial naming practices of the subjects agree in 11 cases out of 14 with the prediction of the UNDERSTAND program, but the program does not predict the adaptive shifts that many subjects made in the course of their problem solving activity from Variable naming to Permanent naming.

In the case of nine of the 14 Monster problems, the pairs of objects in Sentence 3 were listed in an arbitrary order not corresponding to the order of either set of objects. In seven of these nine cases, the subjects introduced some special method to keep track of the monster-globe relationships. In only one of the four cases where objects were ordered from small to large in Sentence 3 did a subject make use of any such special place-keeping devices. The special place-keeping methods were quite varied in nature. One subject assigned numbers to monsters in the order of their mention in Sentence 3. Another subject referred to monsters as occupying “slots,” in order of their current sizes, and as moving from slot to slot. A third represented size by vertical displacement on the page. A fourth (two protocols) reordered the globes by size, thus reformulating Sentence 3. Two subjects kept track of pairings by their position on the instruction sheet, one using his finger for this purpose. Finally, one subject tried, explicitly and unsuccessfully, to use visual imagery to remember the pairings. These renaming methods gave the objects Permanent names, usually associated with spatial position of some kind. The UNDERSTAND program in its present form does not make provision for any reordering processes to conserve short term memory.

CONCLUSIONS

We can draw several conclusions from these findings. First, both the way in which the subject names objects and the way in which he structures the internal problem representation are determined pretty directly by the language in which the problem instructions are written. Subjects initially adopt the naming conventions and representations that follow most directly from a parsing of the instructions. This is precisely what the UNDERSTAND program predicts, as against several alternative strategies mentioned earlier. If the problem is described as a Transfer problem, the subject will represent it as one; if it is described as a Change problem, he will represent it in that way—even though, as we have seen, Change problems are more complex than the isomorphic Transfer problems.

On the other hand, there is some evidence that, as the problem-solving efforts continue, inconvenient naming conventions are abandoned.

As before, there were only three cases of 14 that diverged from expectations, the probability of such an extreme outcome being only .028.
for more convenient conventions (Variable names abandoned for Permanent names). There is also one piece of evidence (the case of L. A.) that a subject may abandon a difficult representation for one that makes solving the problem easier.

Although we have not yet found a way of encoding this particular evidence, perusal of the protocols suggests that the main cue that subjects are responding to in considering a change of naming practice or representation is the load on short-term memory that is imposed by awkward naming or representation, either when it comes to applying the tests of legality of moves or when it comes to recalling the current problem situation, or both. The present version of the UNDERSTAND program is not capable of this kind of adaptive behavior, and will have to be modified to incorporate it if it is to simulate this aspect of the human behavior.

The assumption incorporated in UNDERSTAND that understanding processes will precede solution attempts, without feedback, is partially, but not fully, borne out by the data. A more sophisticated simulation will have to take such feedback into account. Some evidence was found to support the assumption of UNDERSTAND that subjects interpret the instructions of these kinds of tasks by mapping them onto list structures and basic operators drawn from a repertory stored in long-term semantic memory.

REFERENCES


REFERENCE NOTE


(Accepted December 2, 1975)
学霸图书馆 (www.xuebalib.com) 是一个“整合众多图书馆数据库资源，提供一站式文献检索和下载服务”的24小时在线不限IP图书馆。

图书馆致力于便利、促进学习与科研，提供最强文献下载服务。

图书馆导航：

图书馆首页   文献云下载   图书馆入口   外文数据库大全   疑难文献辅助工具