PROGRAM AND PROTOCOL ANALYSIS ON A MENTAL IMAGERY TASK

George W. Baylor
Institut de Psychologie
Universite de Montreal
Montreal, Quebec, Canada

ABSTRACT

This paper presents a LISP 1.5 program and parts of a protocol analysis that model aspects of the behavior on one adult subject (S) solving a set of Guilford's block visualization problems. The model derives from a detailed analysis of S's thinking aloud protocol on four of these problems. Two problem spaces are postulated to account for S's internal representation of the task. Nine operators, evoked by a production system cum goal stack, are used to describe his encoding and problem solving. The program lags the protocol analysis: It respects the two problem spaces but incorporates only five of the nine operators, and these are evoked by hand. The principal problem-solving operators in the image space, Process Block and Tally, both programmed, are described, and the behavior they generate is put in correspondence with, and serves as a set of psychological hypotheses for, S's behavior. Their plausibility rests primarily on the closeness of simulation to S's performance, while their generality has scarcely been tested.

INTRODUCTION

This paper introduces a LISP 1.5 computer program and a collateral protocol analysis that model aspects of the behavior of one adult subject (S) solving some problems from one of Guilford's (14) intelligence tests, Spatial Visualization II. The goal of this research is to try to demonstrate the role of visual mental imagery in the human problem-solving process. The particular task chosen -- a Block Visualization Test (BVT) with visualization instructions -- apparently obliges Ss to construct visual mental images (from memory) in order to be able to represent and solve the verbally presented problems.

The program derives, indeed was constructed from, the detailed analysis of S's 12-minute thinking aloud protocol. Thus, the first part of this paper presents the task and principal results of the protocol analysis. Then the program itself is introduced: first, its way of representing mental images and other non-imagerial, factual information; second, its performance on two test questions, which is compared and contrasted with some stretches of the protocol data it is intended to simulate. Finally, some general conclusions, psychological hypotheses, and directions for future work are proposed.

I. THE TASK AND PROTOCOL ANALYSIS.

A doetoral student in psychology silently reads to himself the following instructions on Guilford's (14) Block Visualization Test (BVT):

> "In this test you will read a description of a solid unpainted block and the manner in which it is painted and then cut into parts. You are to visualize the block, and solve the problems by trying to imagine how the blocks look before and after they are cut up into parts. Do not make any drawings on the test page".

S then turns on the tape recorder and looks at the first test page. On it are four problems, each followed by three or four test questions. He reads and thinks outloud for the duration of the test and, in all, produces a little over 12 minutes of continuous verbal behavior.* The transcribed verbal protocol constitutes the data from which the program was constructed. Parts of it -- Problems BVT #1 and #2 with their first test questions -- are reproduced along the left side of FIGURE 3 in the APPENDIX. (Before going further, the reader is encouraged to turn to FIGURE 3 and to try the two problems for himself.)

Following data analysis techniques largely developed by Newell (21,23) (see also Eastman (9) for an interesting application of this methodology)! this protocol has been extensively analyzed in Baylor (1), which see. Only the main results will be summarized here.

First, the protocol lent itself to the definition of two problem spaces -- an image (I-Space) and a symbolic space (S-Space) -- corresponding to what could be inferred from the protocol about the internal representations and processes S employs on the BVT.

* This is not the place to enter into a discussion of the methodology of protocol collection; indeed, de Groot (6,7) has already done that. In collecting this kind of data, suffice it to emphasize here the importance of a co-operative subject who is experienced at delivering verbal protocols of his thought processes.

Next, the elements of these two problem spaces were used to construct 272 states of knowledge through which S is supposed to have passed in the course of solving the problems. Nine operators and a number of goal-setting processes were postulated to account for the transformations between these states. The sequentiality in S's setting of goals and applying of operators was charted on a problem behavior graph (PBG). In general, the PBG reveals how S accommodates himself to the structure of the task environment (Simon (31)). That is to say, the test questions on the BVT furnish the final states to be attained; the goals are to get these states. Once S has constructed an internal representation of the problem statement, he need only find, apply, and check -- up to some acceptable degree of subjective certainty -- a sequence of actions that achieve the goal states. S clearly relies on the given, desired states in making his selection of an operator in a current state, and his search strategy is another example of means-ends analysis (Duncker (8); Ernst ε Newell (11)).

Finally, a set of 14 productions or condition --> action rules were fit to the regularities on the PBG. The condition sides of the production rules are described in a meta-language of expressions, made up of classes of knowledge or goal statements to get classes of knowledge; the action sides are operators or goals to get desired states.

The production system was then superimposed on the PBG. This shows how the choice of "what to do next" -- be it the act of setting a goal or the application of an operator -- depends both on the well-defined local conditions of the production rules that evoke immediately an action and on a goal stack whereby control reverts to an earlier goal state for selecting the current action. The production system conjoined with the goal stack appears to offer a more flexible way of organizing means-ends analysis than its classic embodiment in GPS (see Newell ε Simon (25); Waterman (34)).

The normal flow of processing, abstracted from S's PBG, is presented in FIGURE 1; it presents the nine operators, the two top level goals, and the classes of objects the operators act on and produce. FIGURE 1 summarizes the three phases or categories of behavior that were identified in the protocol analysis. Recall that the subject does everything in his head, without recourse to paper, pencil, or other external (memory) aids.

First, there is an image construction phase where a representation of the block with its cuts, colors, and dimensions is constructed from the problem statement. This aspect of the BVT requires of a subject that he:

1. (A) construct a mental representation of a block,
   (B) dimensionalized and painted, and
   (C) sliced up into a specified number of little pieces, usually cubes; all this in accord with the description presented in the problem statement.

Second,there is a question and retrieval phase where the test question specifies the desired object. This phase requires of a subject that he:

2. (A) encode the test question, and
   (B) activate an object in memory that best fits the signalment set by the test question.

Two operators were postulated in the protocol analysis to account for S's behavior during these two phases: the encoding and memory retrieval operators, Encode _Input (EI) and Match Schema (MS), respectively. Taken together, they accept as input a sentence (sometimes a clause) and, when successful, they transform it into an I-Space object or schema that is an image representation of the input sentence. S spends nearly half of his total processing time (336/727 sees.) "translating" the problem statements and test questions into internal representations of their meaning.

The rest of his time is devoted to the problem-solving phase, where the 1-Space object activated in Phase 2 is transformed into the sought-after one specified by the test question. This requires of a subject that he:

3. (A) track the distribution of colors (and/or dimensions) from the larger object onto the smaller component parts, a redistribution brought about by the slice transformations, and
   (B) count the number of smaller parts to arrive at the answer to the test question.

* Operators indentified in a protocol analysis at best describe S's behavior, often at a rather gross input/output level. They do not explain it. Programming the operators, that is, rendering their inner working explicit and operational, is the beginning of explanation. However, even my partial attempts at programming EI (and for that matter MS) have added little, if anything, to an explanation of their underlying processes. Nonetheless I shall continue to speak of EI and MS, for their descriptive value, fully aware of the wealth of ignorance this usage implies. See the papers in Minsky (18) for some good approaches to the formidable problems involved in processing natural language.

Of the seven problem-solving operators identified in the protocol analysis, three of them, Process Block (PB), Tally (TAL), and Contradict Block (CB) function exclusively in the I-Space. As can be seen from FIGURE 1, PB takes an I-Space object as input and is evoked by the goal to get cubes or pieces that have the sought-after properties. Its successful application consists in slicing the pieces in question according to the problem specifications and rendering as output a transformed I-Space object with the desired properties.

When the output of PB is the response that no cubes have the desired properties -- the nuTT object -- the operator CB tries to contradict this finding by checking it against known properties of the block. Its output is a confirmation (or disconfirmation) of the response, "None of them."

TAL also takes an I-Space object as input, usually the one delivered by its predecessor PB, and is evoked by the goal to get HOWMANY cubes or pieces have the sought-after properties. Its application usually involves a further slicing of the block and a tally of the exact number of cubes (or pieces) with the desired properties. If this number has already been computed and stored in memory, the operator MS sometimes retrieves the correct number of parts with their I-Space referents. The successful completion of TAL (or MS) supplies an answer to the test question, and S can begin another question or a new problem.

The three operators ADD, SUB, MULT function in S-Space. Like TAL they take as input the current I-Space object and are evoked by the goal to get HOWMANY objects have the desired properties. They perform their arithmetic operations and yield an S-Space object as output. The ninth and final operator is Point_ (PT), a cross-over checking operator that takes the S-Space object as input and "points to" its I-Space referents.*

Note finally in FIGURE 1 that a symbolic operator (SUB, ADD, or MULT) plus the cross-over operator PT presents an alternative to TAL for getting the number of objects in I-Space that have the right properties. In the production system, this was an unresolved production conflict, where the condition sides for determining which "route" to follow were indiscrirainable and the system thus unable to choose between operators.

* See S's behavior at F18-F19 in FIGURE 3, which was encoded as an instance of MULT in the protocol analysis; and F20-F22, encoded as an instance of PT.

II. BASE REPRESENTATIONS IN THE PROGRAM.

The protocol analysis is viewed as a preliminary step in the construction of a program organized into the above three phases. The definition of the problem spaces, and especially of their operators that effect the problem solving, puts a kind of lower bound on the range of symbols that any program must take account of that is put forth as a model of human behavior on this task.

The program is written in LISP 1.5 in the 60 bit words of the 6400/6600 LISP 1.5 of the University of Texas (19) and runs on the CDC-6400 at the Centre de Calcul, Universite de Montreal. The program is much more fully described in (2).

A. The Mental Representation of a Block. In order to "construct a mental representation of a block," what knowledge must a subject possess about the task environment of cubes and blocks and slices and their existence in a three dimensional world? That is, what would seem to be the minimal information he need have stored in long-term memory? and how is this knowledge to be put together?

AI. S-Space. First of all, there is information that holds true about pieces and blocks *in* general. This is called symbolic or S-Space information; it has no specific imagerial referent. For example, all blocks or pieces have FACES; the names of the FACES are (TOP BOTTOM BACK LEFTSDE FRONT RIGHTSDE)*. They also have SIDES: (LEFTSDE FRONT RIGHSDE BACK). FACES have four EDGES, and EDGES have two VERTICES; their names are context dependent -- for example, the TOPEDGE may refer to the TOPEDGE of the FRONT, of the LEFTSDE, of the BACK, or of the RIGHTSDE. Specificity is conferred in the I-Space.

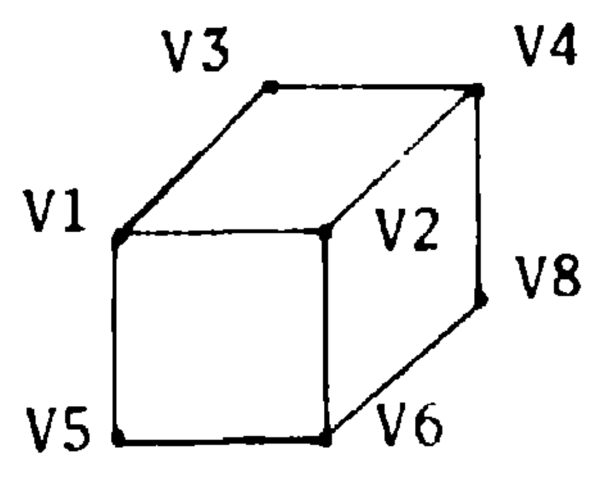
SLICES can be made in blocks in any number of directions. The program only recognizes SLICES oriented in the three dimensions of everyday experience, however, since these are the only ones presented in the BVT: SLICES can be HORIZONTAL, VERTICAL in depth, or laterally VERTICAL. However, when a sequence of slices is generated -- say a set of laterally VERTICAL SLICES -- they must be generated in some order, so the program distinguishes between SLICES generated from left to right (L-R-VERTICAL) and from right to

* The capitalized words like FACES, TOP, BOTTOM, SLICES, HORIZONTAL, etc., are the names of atoms defined in the program; in general, the convention of capitalizing symbols that appear as such in the program will be maintained throughout the description that follows.

left (R-L-VERTICAL). L-R-VERTICAL SLICES, then, refer to lateral cuts generated sequentially from left to right -- like a knife slicing off pats of butter from the left end on the stick to the right. In the program five planar directions are associated with the indicator SLICES: (HORIZONTAL L-R-VERTICAL R-L-VERTICAL B-F-VERTICAL F-B-VERTICAL).*

This is all part of the general information about blocks, pieces, faces, edges, vertices, slices, and their directions that make up the initial S-Space representation of the block. For the atom BLK itself there is a list of eight vertices, CV1 V2 V3 V4 V5 V6 V7 V8), stored under the indicator APVAL; this is a pointer to the I-Space representation of the block.

A2. I-Space. The full image or I-Space representation of a piece or block is a list of eight vertices, called a piecelist; this piecelist is connected to six facelists, each of which is connected to four edgelists, each of which is connected to two vertices. From each of the (eight) vertices, moreover, there are (at least three) directional pointers to neighboring vertices along the three spatial dimensions: UP-DOWN; RIGHT-LEFT; TOWARDS-AWAY. This structure, partially shown in FIGURE 2 is a one-way graph structure with 48 possible pathways from the piecelist, through the six facelists and twelve edgelists, to the eight component vertices. Between vertices there are at least 24 additional connections. The piecelist, (VI V2 V3 V4 V5 V6 V7 V8), is called the base block, and the following right side projection is one of its possible pictorial representations (with the eight vertices labelled as they are used consistently throughout the program):



* The VERTICAL in depth SLICES, B-F-VERTICAL and F-B-VERTICAL, can be made from BACK to FRONT or from FRONT to BACK, respectively. The subject's behavior varied. For the HORIZONTAL SLICES, however, S always generated them from the TOP to the BOTTOM (and never from the BOTTOM up) so HORIZONTAL SLICES are understood to be in the T-B-HORIZONTAL direction (with the sixth, logically possible direction never appearing). Slice directions are emphasized here since the order of generation is viewed as an aspect of S's behavior to be simulated as well as a psychological parameter to help account for individual differences.

This way of representing a block is considerably more redundant than, for example,Guzman's (15), which like most scene analysis programs (see Rosenfeld (29)) makes use of a co-ordinate system for localizing and measuring distances between vertices. While such a co-ordinate system greatly facilitates computation, it was explicitly eschewed here as being too "strong" an assumption to include in a model of human visual imagery.

111. HUMAN AND MACHINE PROTOCOLS.

The program's behavior, or pictorial surrogates thereof, on BVT Problems #1 and #2, each with its first test question, is presented along the right side of FIGURE 3 in the APPENDIX. The pictures preserve the orthographic projections of the mental images the subject said he was using -- a top projection for BVT #1 and a front/right side projection for BVT #2. These two examples will be used to illustrate the functioning of the program's operators, whose names, also on the right side of the page, are entered under the heading EVALQUOTE CARDS. The transcribed protocol of S's behavior is presented on the left side of the page, with his processing times for speech units and pauses (≥ 0.5 sees.) in the extreme left column.

During Phase 1 the program encodes the problem statement by means of the routines DIMENSIONALIZL; PAINT, CUT-INTO (or CUT-UP); these routines are the operational specifications of the operators, El and MS, identified in the protocol analysis. As can be seen from the first example (BVT #1) in FIGURE 3, these routines serve to particularize a general block by annexing descriptive information (1) to the property lists of the affected facelists and (2) to the piecelist of the base block itself. PAINT assigns the attribute HAVE to the base block with the'fact" (in S-Space) as value that the block has both red and blue faces. To this fact, moreover, it associates the indicator I-SPACE with the appropriate facelists as value -- to ensure the crossover from the fact to its imagerial referents.

This double assignment of information has to do with the observable dual capacity of humans to know as a fact that the block, once painted, has red sides and a blue top and bottom, without having to actually generate up the painted mental image for the six (successive) faces -- but to be able to do so if necessary. This sort of quick/slow memory system receives considerable empirical support from Paivio's (26) and Ernest & Paivio's (10) work on Ss' responses to word association tests. See also Bergson (5) on 1'effort intellectuel.

As to S's behavior during Phase 1, it is fairly clear that he only arrives at his image representation of a "1 inch by 4 inch by 4 inch block" by putting it together from its component faces, whose images he evidently dimensionalizes at A12-A13: "the faces are two 4 by 4 squares and the ah little parts at 1 by 4 rectangles." After that he proceeds to paint the block, as specified, without difficulty.

After the block has been constructed, dimensionalized, and painted, it is then cut into some number of pieces. There are two routines in the program that translate the slice instructions: CUT-INTO when the block is cut into some specified number of CUBES (as in BVT #1), and CUT-UP when the block is cut into PIECES (as in BVT *2).

For example, in BVT #, CUT-INTO (BLK 16 ONE-INCH CUBES), encodes the fact that the block has sixteen 1 inch cubes as PARTS: more importantly, it figures out and stores how many have to be made in the block, and in what direction, in order to arrive at the designated number of cubes In this respect, CUT-INTO is a kind of feasibility test.

In BVT #2 there are three instances of CUT-UP (E4-E7). Like CUT-INTO each application of CUT-UP assigns to the piecelist of the base block (1) the SLICE directions, (2) the number of times each is to be applied, and (3) the factual information about the slice that localizes it on the block. In addition, CUT-UP constructs the I-Space representations of the SLICES.

Now a slice is nothing but a plane that is passed through an object in a certain direction; when this object is a parallelepiped such a plane has the effect of creating four new vertices (unless, of course, the slicing plane passes through already extant vertices). In the program, par consequence, a slice is represented in I-Space the same way a face is: as a list of four vertices. On the base block a new I-Space slice is stored with its direction under the indicator I-SPACE. Finally, in order not to forget the locations of these new vertices, each is inserted in a directional network of vertices that is maintained on the base block.

While little effort was made to put the machine's behavior in close correspondence with the subject's during this phase, the data structures the two construct for representing the problem statements appear highly similar.

During Phase 2 ENCODE-QUESTION translates the test question into a S-Space schema associated with OBJ_0; this is the source of goals for subsequent processing: to get all of OBJ_0 completed in I-Space.

Next, MATCH-SCHEMA retrieves and activates some I-Space object with which to begin problem solving. In BVT #1 both the subject and program recover and activate the "red and blue faces" whose I-Space representations become the ACTIVE-ELEMENTS and whose properties become the sought-after ACTIVE-ATTRIBS: (2 ((COLOR BLUE) (COLOR RED))) on the sliced up block. On BVT #2, there is nothing much to retrieve and problem solving begins with the base block itself.

It is during Phase 3, the problem-solving phase that PB and TAL generate by far the most interesting behavior in the protocol. One example of PB and three of TAL will be presented.

During the protocol analysis S's behavior from B4 to B6 (see FIGURE 3) was identified as an instance of PB. S expresses the input object at B4: "Ah, the sides are red and the top and bottom are blue"; the slice transformations at B5: "Cut them that way"; and the output object at B6: "Well it would be all of them on the border".

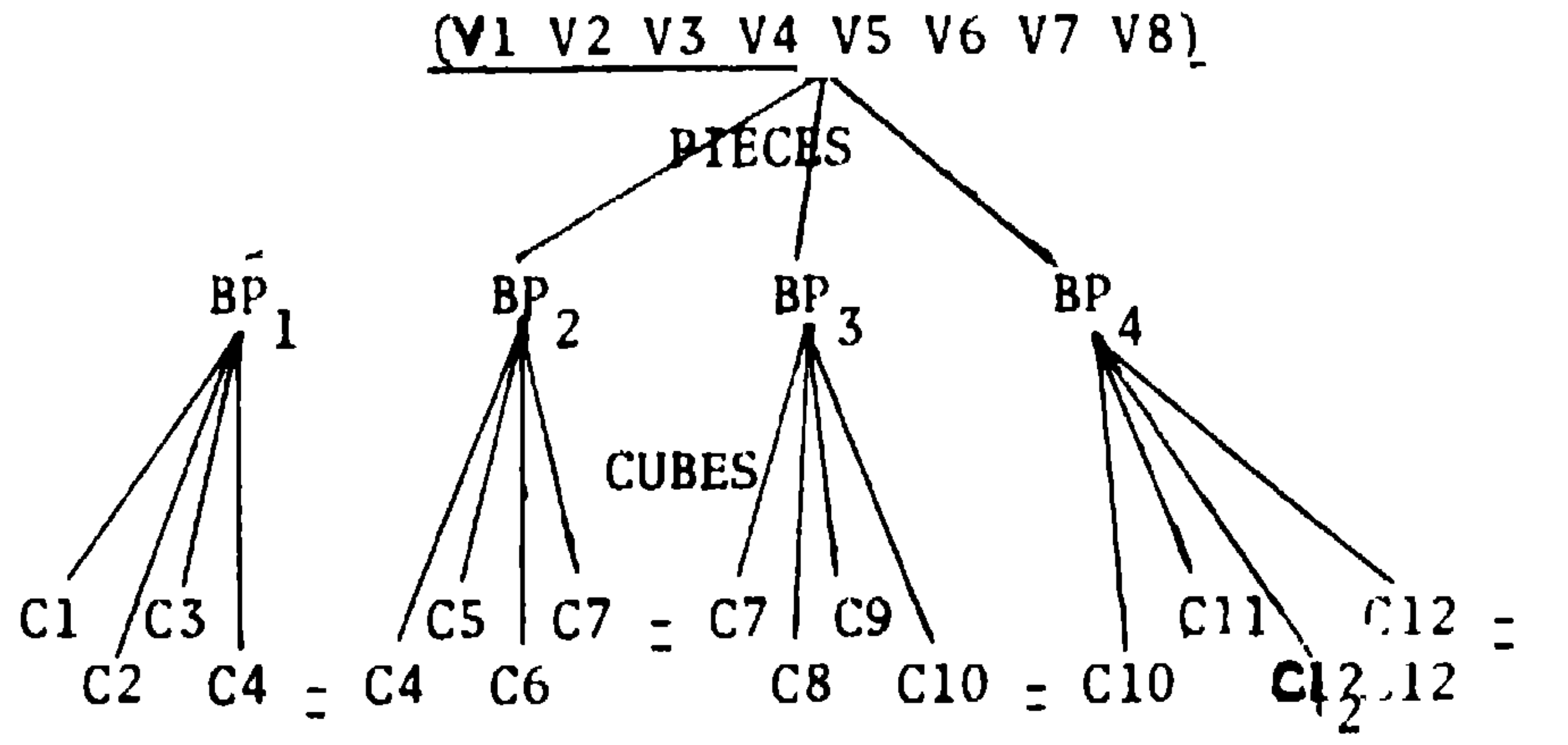
How does the PB of the program explain this behavior? First, PB slices the base block, (VI V2 V3 V4 V5 V6 V7 V8), the number of times previewed by CUT-INTO at A17-A18 above: three times in the L-R-VERTICAL direction and then three times in the B-F-VERTICAL direction, without taking account of interactions between the two sets of SLICES. Second, while the program is constructing the eight new PIECES, it also tracks the active attributes: (2 ((COLOR BLUE) (COLOR RED))), activated by MATCH-SCHEMA at B3-B4 above. This yields four ACTIVE PIECES with the sought-after properties. Third, PB chunks and calls them BORDER-PIECES, a category defined by IMAGE (S-BLOCK) at AI above. FIGURE 3A is a printout of the actual machine representation of the BORDER-PIECES, whose pictures appear opposite B6 on the protocol. In short, PB proceeds heuristically by making an initial slice in the block that ignores the interaction effects between slices in the hope of finding some global, geometric partitioning of the cut up block that meets the requirements set by the test question.

How does the operator TAL explain the rest of S's behavior? The goal statement that evokes it is quite explicit at B7-B9: "Ah, so that's how many on the border?" S then begins to count cubes and says at BII: "One, two, three, four." From the way his speech is segmented, moreover, it seems reasonable to assume that he generates each border piece in turn, forming cubes from each with the slice set in the other direction (B11-B14). He avoids the trap of double counting the corners, which is, presumably, what his utterance at B10 signals: "On the periphery?" That is, he will count along the top "periphery" where the identity of corner cubes is more easily recognized than along the sides "on the border" (B9) where there are, in fact, 16 red faces.

The pictorial representation of the behavior that the program's TAL generates is opposite B11-B14 in FIGURE 3. For each BORDER-PIECE TAL retrieves the SLICE directions in the other direction that have yet to be made. Starting with the left most BORDER-PIECE, (VI V3 V5 V7 64 65 66 67), the three B-F-VERTICAL SLICES are intersected to create four new CUBES. TAL creates two new inside vertices for each new CUBE it constructs.

TAL now generates the next BORDER-PIECE, (396 397 398 399 V1 V2 V5 V6), then retrieves the L-R-VERTICAL SLICES, and slices this BORDER-PIECE three times -- and so on around the border. The program must be careful to avoid the trap of double counting the two copies of the corner CUBES, each of which issues from different generators. This means testing for their identity. The routine which does this is viewed as a psychological parameter of the program since its inclusion or exclusion determines whether or not the program will avoid the erker of double counting -- a "good" error that many people, especially children, are prone to commt on *tksne* proh-enes.

Together, PB and TAL create the following tree structure for BVT #1, Question #1 (where - signs signal identities between'CUBES and the shorthand symbols BP_{1-4} and C_{1-12} are substituted for the lengthy vertex lists of the BORDER-PIECES and CUBES, respectively). Since the CUBES are parts of the block as well as being parts of the BORDER-PIECES, the list of CUBES is ultimately associated directly to the piecelist of the base block:

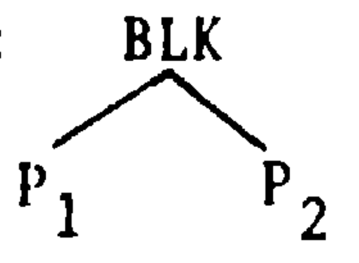


* From the point of view of unambiguously interpreting the protocol at least one other parallel record of S's sequential behavior should accompany his verbal report, be it a sequence of switch settings (Laughery ¢ Gregg (17)), eye move movements (Winikoff (35); Newell (22)), or a luminous finger in a darkened room that the subject uses to trace and exteriorize his sequence of mental images (Racine (27)).

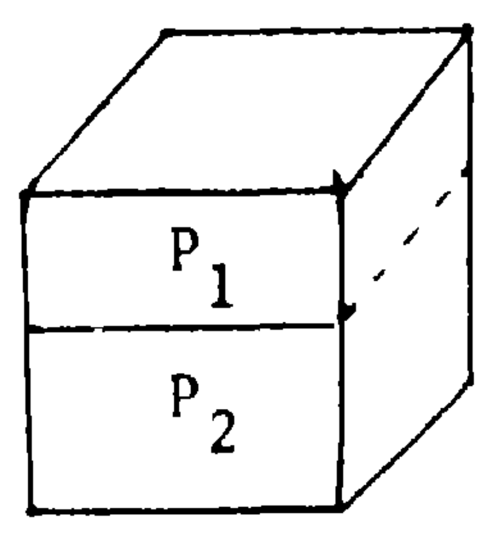
The program's simulation of BVT #1, Question #1, is a rather close fit to S's behavior. To some extent this is because the program was constructed from this example, but unlike just any old program that can take blocks and cut them up, the emphasis here was on the exact reproduction of S's behavior. In the absence of contradictory evidence, both the vertex list representations of the mental images and PB, TAL, and associated processes for extracting and creating information are regarded as functionally equivalent to S's mental images and imaging activity, respectively.

The second example of TAL is more revealing since its first application fails. The first question of BVT #2 is "4. How many pieces are there?" MATCH-SCHEMA retrieves the base block itself and this calls up the operator: TAL (BLK), a computaticciulrymore complex argument for TAL than the above BORDER-PIECES.

TAL is a recursive routine; its outputs at one level become its input arguments at the next. In this example, TAL (BLK) first retrieves the slice directions that are to be made in the block: (HORIZONTAL R-L-VERTICAL F-B-VERTICAL) as set by CUT-UP at E4-E7 above. This list of three slice directions implies three applications of TAL as well as setting the terminal condition for TAL's recursion. For each direction TAL calls the appropriate slice routines to cut the piece the requisite number of times in that direction. For this a FOCUS FACE is necessary; hence, CSET (FOCI FRONT) since the effects of the HORIZONTAL SLICE, the first to be made, are only "visible" on the FRONT or RIGHTSDE. The first level piecetree TAL constructs is simply:

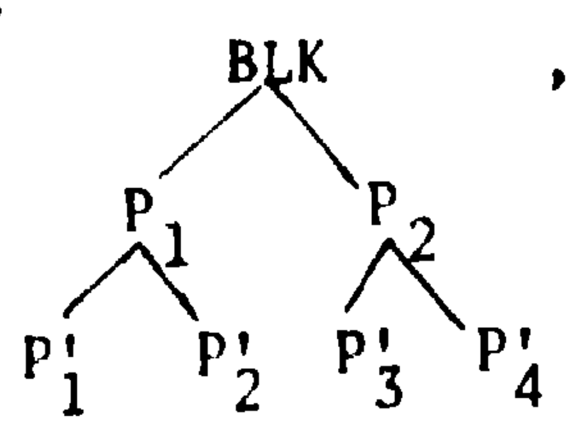


corresponding to the picture:

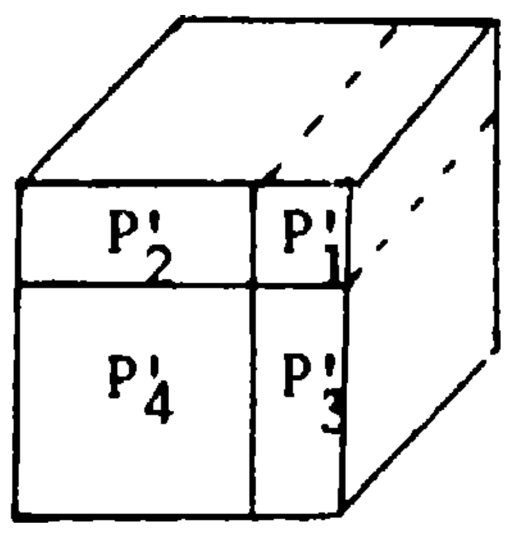


at F2 on the protocol of FIGURE 3.

Next, TAL((P1 P2)) slices each piece in the R-L-VERTICAL direction, yielding the following piecetree:



, corresponding to



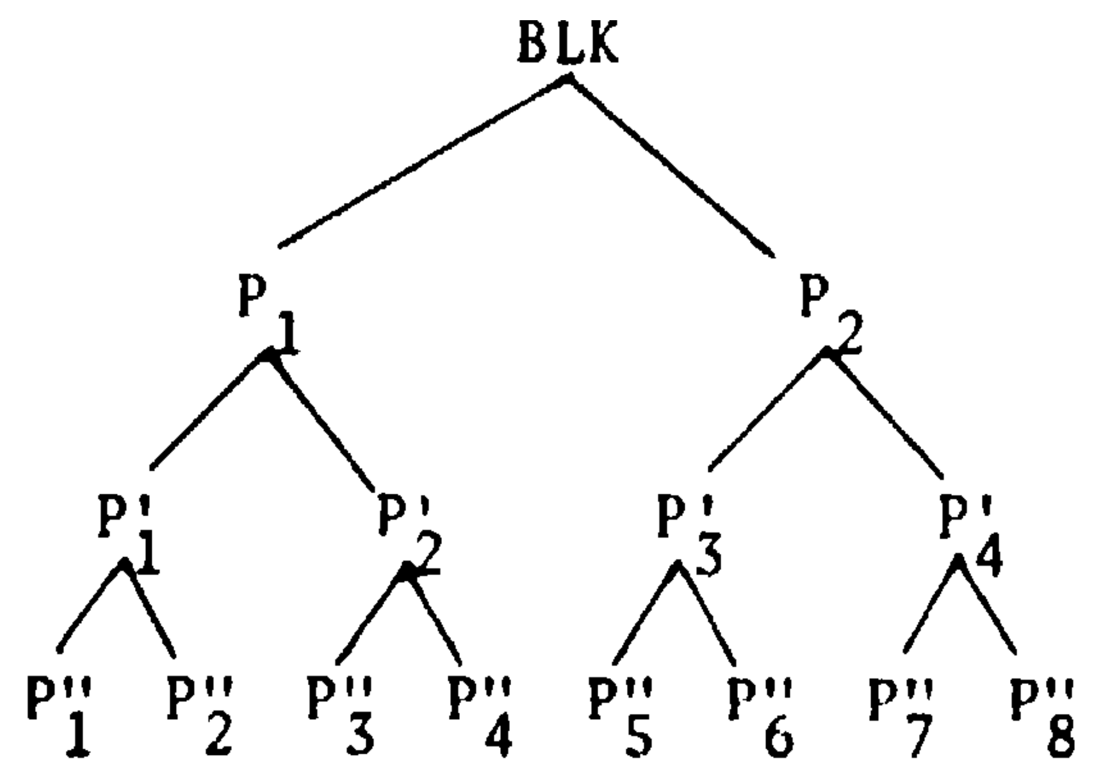
at F5 on the protocol

TAL next tries to compute the effects of the F-B-VERTICAL SLICE: TAL ((P'$_1$P'$_2$ P'$_3$ P'$_4$)). It fails because it finds the TOP of the first piece it tries, P$_1$' to be DISEMBODIED. This (unexpected) failure of the program corresponds exactly to S's failure:

F6-F8: "And a third one inch from the front
Uhm.
Oh gosh I find that very hard."

Why are the TOP's of the pieceset (P' P'$_1$ P'$_2$ P'$_3$P'$_4$ DISEMBODIED? It is because when a new piece is constructed, only three of its FACES are fully developed, in contrast with the six FACES in the full representation of the base blcck, as presented in FIGURE 2. For PI, for example, its three full-bodied FACES are its RIGHTSDE and LEFTSDE -- its bounding planes formed by the block's RIGHTSDE and the R-L-VERTICAL SLICE -- and its FRONT -- the FOCUS FACE from which this PIECE is viewed. Since, however, the FOCUS on which the effects of a third, F-B-VERTICAL, SLICE can be read is assumed to be the TOP of PI -- which is DISEMBODIED -- the program is unable to slice further PI. Such a partial representation of a new piece in I-Space constitutes a heuristic definition, and it is proposed as a functionally equivalent structure to S's mental images,which helps to account for his failure at F6-F8.

So the FOCUS FACE is changed (opposite F9 on FIGURE 3) to: CSET (FOCI TOP), and TAL (BLK) is reapplied. The sequence of images that this repetition of TAL (BLK) generates is shown pictorially on FIGURE 3, starting at F9 and ending with the picture at F20-F22, corresponding to the following piecetree:



Note that in reapplying the first HORIZONTAL SLICE that the shift of FOCUS FACE from the FRONT to the TOP causes no problem since the old, already constructed PIECES' P$_1$ and P$_2$, are retrieved; they already have FRONT FACES, so the fourth full-bodied FACE, the TOP, need only be added to their partial representations.

From F18-F22 there is obviously a discrepancy between the program's and S's behavior; namely, in the protocol analysis S was postulated to apply the symbolic operator MULT at F17-F19: "and that just doubles it," followed by PT at F20-F22, where he elaborates just the front four of the eight pieces, (P"$_1$ P"$_3$ P"$_5$ P"$_7$). This to stretch must be counted as an error of simulation. It can perhaps be remedied once the MULT and PT sequence have been programmed.

Finally, since there are no more SLICES TAL assigns the eight terminal PIECES to the piece-list of the base block (see FIGURE 3B) and completes the object schema: OBJ$_0$
HOW-MANY: 8.

En resume, a number of routines, which are entered at the EVALQUOTE level in the program, have been constructed to respond to the BVT task demands. First, during Phase 1 the routines DIMENSIONALIZE, PAINT, CUT-INTO or CUT-UP serve to encode the problem statement. Second, during Phase 2 the operators ENCODE-QUESTION and MATCH-SCHEMA are used to encode the questions posed on the BVT and to retrieve appropriate objects with which to commence problem solving. Finally, during Phase 3 the operators PT, PB, and TAL are employed to construct and point to the I-Space objects described by the test questions. The accent has been put on the I-Space problem-solving operators, PB and TAL; the third one identified in the protocol analysis, CB, has not yet been programmed, nor have the three symbolic operators, MULT, SUB, ADD. For all three phases the program's behavior has been put in correspondence with the human's on two example problems and test questions.

IV. CONCLUSIONS

From the point of view of artificial intelligence this work can be regarded as a strategy for program construction, while from the viewpoint of cognitive psychology, as an operationalization of concepts and a source of psychological hypotheses (Frijda (12); Hunt (16); Reitman (28)). It would be nice to construct more crossover links that span the two (Newell (24)).

First, to workers in "pure" artificial intelligence the task must seem trivial, as indeed it is compared to proving theorems, in the predicate calculus, solving tough symbolic integration problems (Slagle (32); Moses (20)), or playing good chess (Baylor Simon (4); Greenblatt, Eastlake ¢ Crocker (13)). Moreover, the program has not yet been demonstrated to be very powerful, in that its performance on only two test questions has been presented. Nor is there any evidence of its generality, either on a wider, expanded range of BVT problems, or on the full battery of intelligence tests that Guilford (14) has designed to measure what he calls Cognitive Figural Transformations in his structure of intellect factor analytic model.

No, if it be of interest to workers in AI, it is likely to be because of its approach to the problem of representation. The vertex list notation and operators used to represent and process mental images do seem to capture at least some of the right properties of the few geometric figures they deal with (see also, Simon (30)). Of course, this is not surprising since this notation is but an adaptation of parts of plane and solid geometry, a mathematics devised for expressing just such features of the Euclidean world. This representation also has a certain cumbrousness about it and does not seem to program readily in a list-processing language like LISP, at least not without creating special routines for associating descriptive information to the vertex lists*; in this respect, indeed, it would probably be better to program the mechanics of the image space closer to the hardware level, perhaps along the lines initiated by Sutherland (33). Certainly much work needs still to be done on the "perceptual languages" for encoding the visual world and on the languages of memory and mental imagery for calling it back up; how, for example, are dreams to be represented in a computer, probably the most pervasive source of visual mental imagery in humans?

Second, to cognitive psychologists this work is presented as a preliminary model of how humans make use of mental imagery in solving block visualization problems; said otherwise, it is a first attempt at operationalizing the factor Guilford (14) calls Cognitive Figural Transformations. The program has furnished a number of non-obvious psychological hypotheses, also. For example, the way in which the program constructs and heuristically defines a partial representation of a new piece with a single, usually "visible," FOCUS face that stands for the whole proves insufficient when TAL must "read off" the effects of interacting slices on a second (disembodied) face of the same piece. This was offered as part of the explanation of S's failure.

As a second example, it was also proposed that the errors of double counting that people sometimes make on these problems comes from just their failure to recognize or identify two faces generated in different contexts as belonging the the same piece. Such a test for identity was offered as a psychological parameter of the program in that it can be used to predict differentially the commission or omission of double counting errors. This signals, moreover, at least one important difference between visual mental imagery and visual perception: unlike objects in the visual field mental images of the "same" object produced at two different points in time do not remain available to the information processor for inspect (or identification) at will. (The psychological aspects of the program are developed more fully in Baylor (3).)

And the view from the bridge that spans artificial intelligence and cognitive psychology? On the one side there is the obvious need for bigger and better programs that are more sensitive to the data of human information processors, and on the other side, there is a corresponding need for experimental support that puts to the test the psychological implica tions of these programs as models. Two way traffic between the two cannot help but further our understanding in both fields.

ACKNOWLEDGEMENTS

This paper summarizes parts of a doctoral dissertation at Carnegie-Mellon University. I would particularly like to express my gratitude to Dr. Herbert A. Simon for his judicious counsel and continued support over a number of years. Also, my thanks to the referees of the Conference Committee for their helpful comments in revising this paper; and to Mmes F. Lasry and Dufresne in typing parts of this paper. Finally, a word of thanks to Mile Tamara Lemerise for her help in preparing the text. This work has been financed by the Ministere d'Education of Quebec.

* While describing the program in the preceding section, no mention was made of the machinery required to associate descriptive information to (and retrieve it from) lists. Atomic symbols have unique addresses in LISP, but two "identical" lists, i.e., two lists containing identical ordered sets of symbols, point to different addresses. Consequently, the function DEFLIST, which creates property lists on atoms, could not be used for associating the attributive information to the piecelists, facelists, etc. A new, comparatively slow function called GEOLIST, among others, had (Cont'd) to be written. All told, the program runs to over 1500 LISP instructions. Along with the fact that the program runs on a LISP interpreter system, these factors help to explain the program's relative slowness: It took on the order of 300 sees, on BVT #1 with Question #1 and 140 sees, on BVT #2 with Question #4.

REFERENCES

1. Baylor, G.W. A treatise on the mind's eye. I. Methodology and protocol analysis. Montreal: Universite de Montreal (Working paper MCP #1), 1969.

2. Baylor, G.W. A treatise on the mind's eye. II. Program structure and performance. Montreal: Universite de Montreal (Working paper MCP #4), 1971.

3. Baylor, G.W. A treatise on the mind's eye. III. Psychological interpretation. Montreal: Universite de Montreal (Working paper MCP #6), 1971.

4. Baylor, G.W. ε Simon, H.A. A chess mating combinations program. AFIPS Procs. 28. Washington: Spartan, 1966. pp. 431-47.

5. Bergson, H. L'effort intellectuel. Revue philosophique, 53, 1-27, 1902.

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

7. de Groot, A.D. Methodology. The Hague: Mouton, 1969.

8. Duncker, K. On problem solving. Psychol. Monographs, 58, No. 270, 1945.

9. Eastman, CM. Cognitive Processes and ill-defined problems: A case study from design. In D.E. Walker £ L.M. Norton (Eds.), Procs. 1JCA1. Bedford, Mass.: The MITRL Corp., 1969. pp. 669-90.

10. Ernest, Carole H. ε Paivio, A. Imagery and verbal associative latencies as a function of imagery ability. Can. J_. of Psychol., 25, 1971, 83-90.

11. Ernst, G.W.€ Newell, A. GPS: A case study in generality and problem solving. New York: Academic Press, 1969.

12. Frijda, N.H. Problems of computer simulation, Behav. Sci., 12, 1967, 59-67.

13. Greenblatt, R.D., Eastlake, D.E. III ε Crocker, S.D. The Greenblatt chess program. Procs. FJCC, 1967, 801-10.

14. Guilford, J.P. The nature of human intelligence. New York: McGraw-Hill, 1967.

15. Guzman, A. Decomposition of a visual scene into three-dimensional bodies. Procs. FJCC, 1968, 291-304.

16. Hunt, E. Computer simulation: Artificial intelligence studies and their relevance to psychology. Ann. Rev, of Psychol., 19, 1968, 135-68.

17. Laughery, K.R. ε Gregg, L.W. Simulation of human problem-solving behavior. Psychometrika, 27,, 1962, 265-82.

18. Minsky, M. (Ed.) Semantic information processing. Cambridge, Mass.: The MIT Press 1968.

19. Morris, J.B. ε Singleton, D.J. The University of Texas 6400/6600 LISP 1.5, an adaptation of MIT LISP 1.5 Austin: The University of Texas (ditto ed), 1968.

20. Moses, J. Symbolic integration. Unpub. Ph.D. dissertation. Cambridge, Mass.: Mass. Inst, of Technology, 1967.

21. Newell, A. Studies in problem solving: Subject 3 on the crypt-arithmetic task DONALD + GERALD = ROBERT. Pittsburg, Pa.: Carnegie Inst, of Technology, 1967.

22. Newell, A. Eye movements and problem solving Computer Science Research Review. Pittsburg Pa.: Carnegie-Mellon Univ., 1967, 28-40.

23. Newell, A. On the analysis of human problem solving protocols. In J.C. Gardin ε B. Jaulin (Eds.), Calcul et formalisation dans les sciences de l'homme. Paris: CNRC, 1968. pp. 145-85.

24. Newell, A. Remarks on the relationship between artificial intelligence and cognitive psychology. In R. Banerji ε D. Mesarovic (Eds.), Theoretical approaches to non-numerical problem solving, 28, Berlin: Springer-Verlag, 1970. pp. 363-400

25. Newell, A. ε Simon, H.A. Human problem solving. New York: Prentice-Hall, in press

26. Paivio, A. Mental imagery in associative learning and memory. Psychol. Rev., 76, 241-63.

27. Racine, B. La transformation de 1'image mentale. Unpub. M.A. thesis. Montreal: Universite de Montreal, 1971.

28. Reitman, W.R. Cognition and thought. New-York: Wiley, 1965.

29. Rosenfeld, A. Picture processing by computer. Computing Surveys, 1, 1969, 147-76.

30. Simon, H.A. An information-processing explanation of some perceptual phenomena. Brit. J. of Psychol., 58, 1-12.

31. Simon, H.A. The sciences of the artificial Cambridge, Mass.: The MIT Press, 1969.

32. Slagle, J.R. A heuristic program that solves symbolic integration problems in feshman calculus. In E.A. Feigenbaum ε J. Feldman (Eds.), Computers and thought. New York: McGraw-Hill, 1963. pp. 191-203.

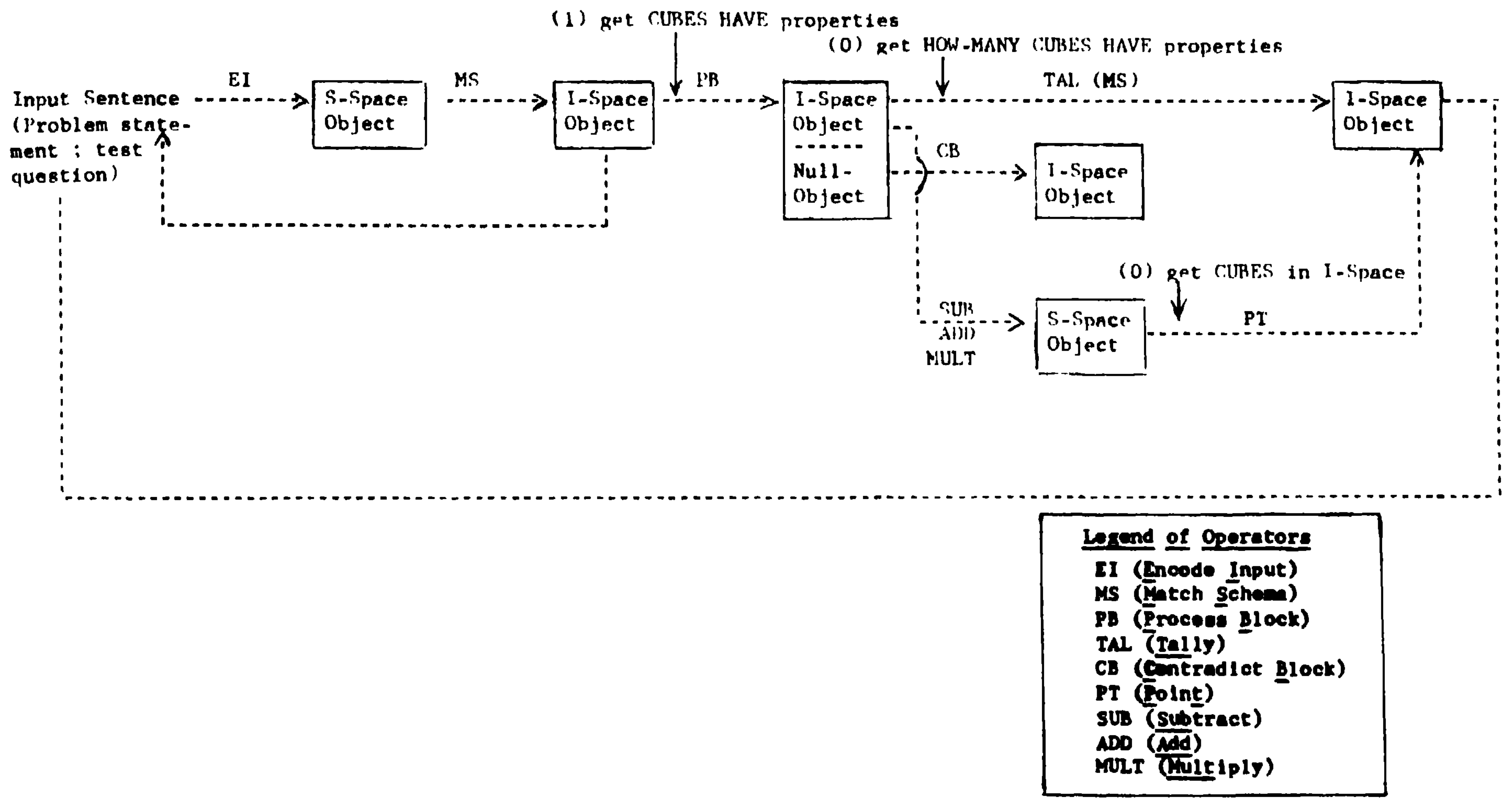
33. Sutherland, I.E. Sketchpad: A man-machine graphical communication system. Lexington, Mass.: Lincoln Lab. (Tech. Rep. No. 296), 1963.

34. Waterman, D.A. Generalization learning techniques for automating the learning of heuristics. Artificial Intelligence, 1, 121-70.

35. Winikoff, A. Eye movements as an aid to protocol analysis of problem solving behavior. Unpub. Ph.D, dissertation. Pittsburgh, Pa.: Carnegie-Mellon Univ., 1967.

APPENDIX

Figure 1 : "Normal" flow of processing in the protocol.



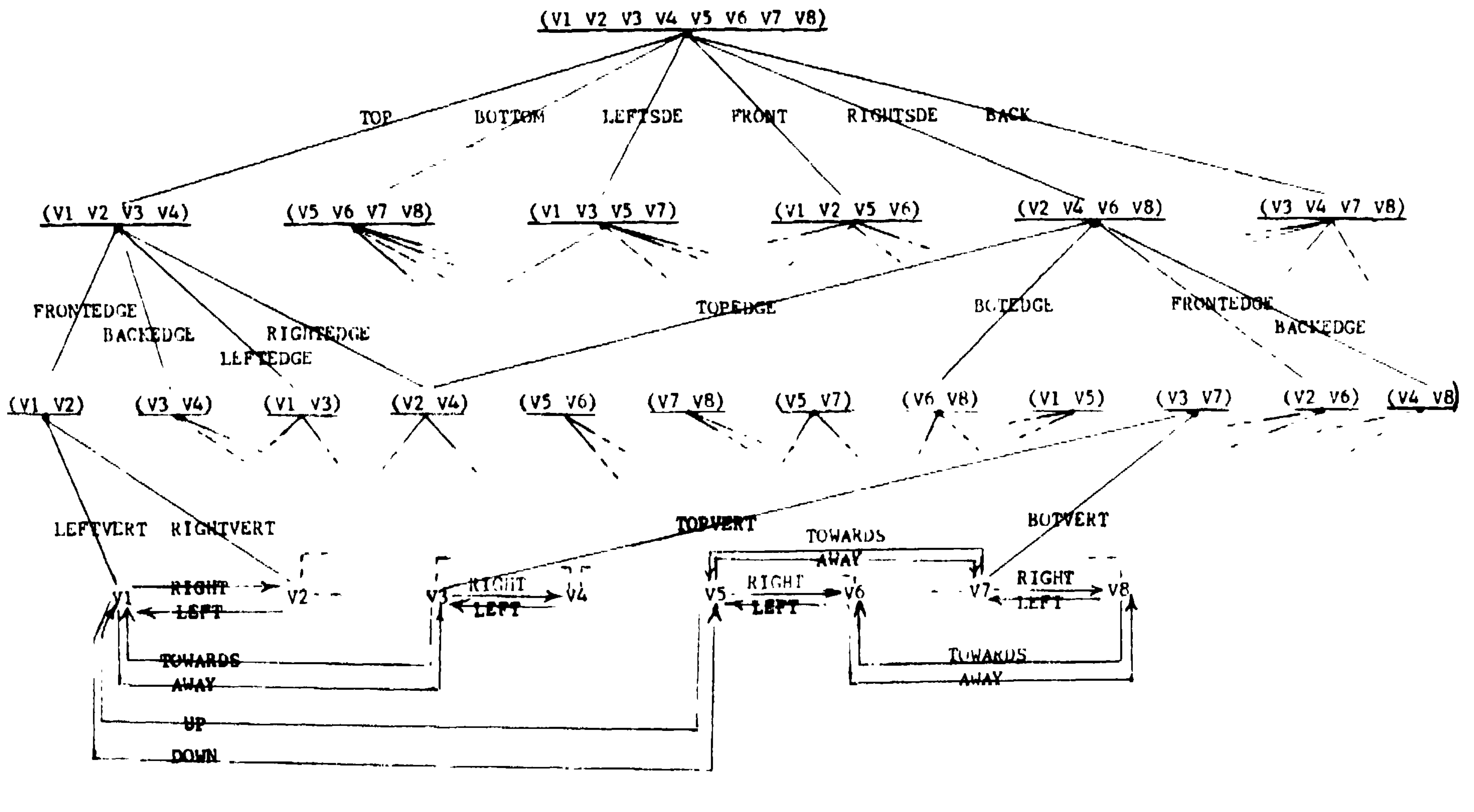


FIGURE 2: Partial I-Spaca representation of the base place, (V1 V2 V3 V4 V5 V6 V7 V8).

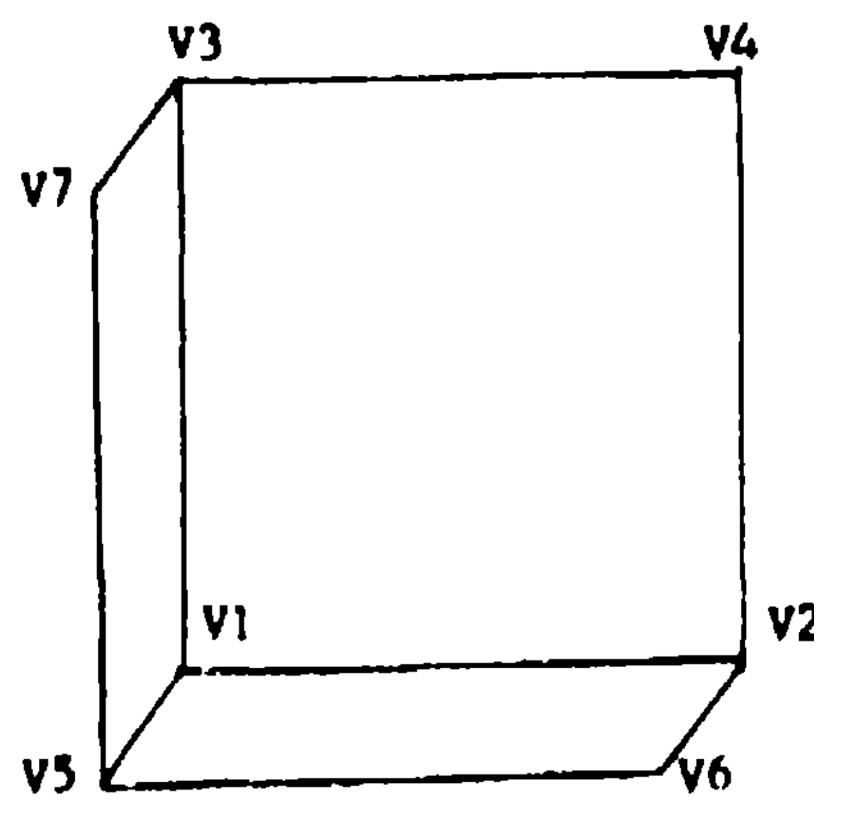
BVT #1: "The four narrow sides of a 1 inch by 4 inch by 4 inch block are painted red. The top and bottom are painted blue. The block is then cut into sixteen 1 inch cubes.
1. **How many cubes have both red and blue faces?**
2. **How many cubes have one red and two blue faces?**
3. **How many cubes have no painted faces?"**

Secs.	Protocol
1.7	A1: "All right, the first, sorry,
1.1	(Pause.)
4.7	A2: The four narrow sides of a (phone rings) A3: 1 inch by 4 inch by 4 inch block are painted red. Uhh.
1.8	
1.0	A4: Answer the phone. All right, try again:
1.1	
5.5	A5: Ah, the four narrow sides of a 1 inch by 4 inch by 4 inch block,
1.1	
2.5	A6: 1 inch by 4 inch by 4 inch block.
2.0	
1.3	A7: Do I have that visualized?
1.8	
2.6	A8: 1 inch by 4 inch by 4 inch, I see.
0.7	
0.8	A9: No, I don't see.

EVALQUOTE CARDS FOR PROBLEM NUMBER 1

PRINTOUT OF TOP LEVEL PROGRAM STRUCTURE (OR PICTORIAL REPRESENTATIONS OF ITS OUTPUT)

```
IMAGE (S-BLOCK)
      0       0
   BLK
     APVAL:   (V1 V2 V3 V4 V5 V6 V7 V8)
     SLICES:  (HORIZONTAL L-R-VERTICAL R-L-VERTICAL
               B-F-VERTICAL F-B-VERTICAL)
     PIECES:  (BORDER-PIECES CENTER-PIECES CORNERS CUBES)
     SIDES:   (LEFTSDE FRONT RIGHTSDE BACK)
     FACES1:  (TOP BOTTOM BACK LEFTSDE FRONT RIGHTSDE)
     FACES:   (TOP BOTTOM LEFTSDE FRONT RIGHTSDE BACK)

IMAGE (1-BLOCK)

   (V1 V2 V3 V4 V5 V6 V7 V8)
     TOP:   (V1 V2 V3 V4)
     BOTTOM:   (V5 V6 V7 V8)
     LEFTSDE:   (V1 V3 V5 V7)
     FRONT:   (V1 V2 V5 V6)
     RIGHTSDE:   (V2 V4 V6 V8)
     BACK:   (V3 V4 V7 V8)
     NAME:   BLK
     PIECES:   ((V1 V2 V3 V4 V5 V6 V7 V8))

DIMENSIONALIZE (BLK 1 4 4)
               0        0

   (V1 V2 V3 V4 V5 V6 V7 V8)
     DEPTH: 4
     WIDTH: 4
     HEIGHT: 1
     THREE-D: (1 4 4)
```

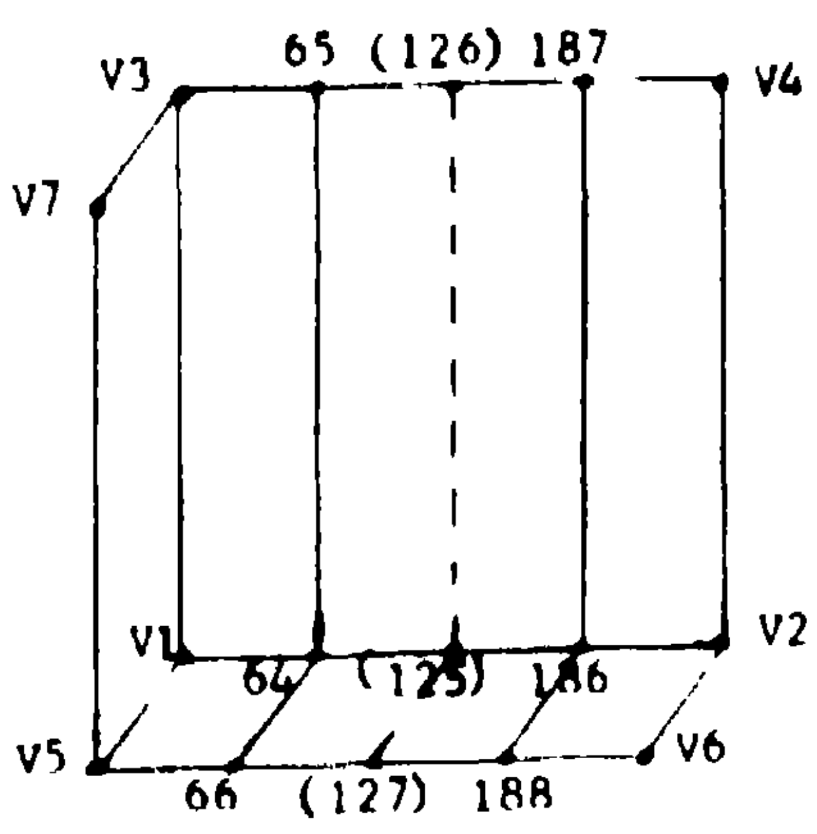


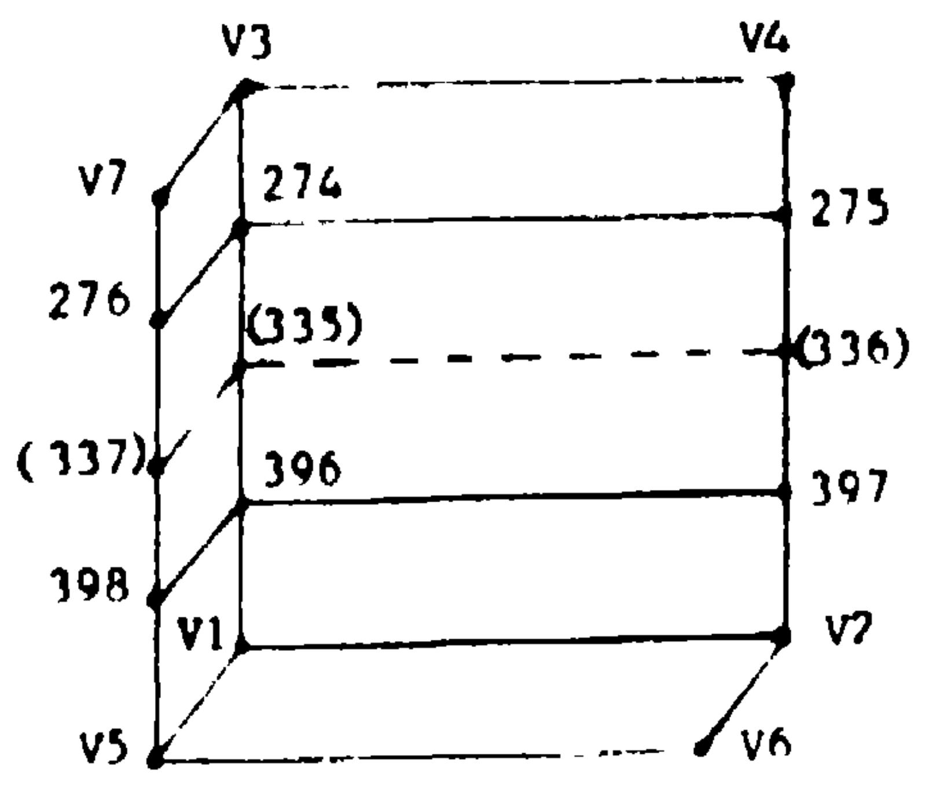
2.5	
1.2	A10: 1 inch by 4 inch.
3.8	
1.8	A11: Okay, so the face of the
1.6	
3.5	A12: the faces are two 4 by 4 squares and the uh
2.2	
4.8	A13: little parts are 1 by 4 rectangles, okay. The four narrow sides
1.0	
1.1	A14: are painted red.
2.3	
1.7	A15: The top and bottom are painted blue.
1.5	
0.6	A16: All right.
1.9	
2.5	A17: The block is then cut into sixteen 1 inch cubes.
1.4	
0.7	A18: All right.
0.9	
2.0	B1: (1) How many cubes have both red and blue faces?
2.0	
1.5	B2: Well, God.

```
    (V1 V2 V3 V4)      (V5 V6 V7 V8)
      DEPTH: 4           DEPTH: 4
      WIDTH: 4           WIDTH: 4
      TWO-D: (4 4)       TWO-D: (4 4)
      NAME: TOP          NAME: BOTTOM

    (V1 V3 V5 V7)      (V1 V2 V5 V6)      'V2 V4 V6 V8)      (V3 V4 V7 V8)
      DEPTH: 4           WIDTH: 4           DEPTH: 4           WIDTH: 4
      HEIGHT: 1          HEIGHT: 1          HEIGHT: 1          HEIGHT: 1
      TWO-D: (1 4)       TWO-D: (1 4)       TWO-D: (1 4)       TWO-D: (1 4)
      NAME: LEFTSIDE     NAME: FRONT        NAME: RIGHTSIDE    NAME: BACK

PAINT (BLK SIDES RED)
      1            0
    (V1 V2 V3 V4 V5 V6 V7 V8)
      HAVE: (SIDES COLOR RED)

    (SIDES COLOR RED)
      I-SPACE: ((V1 V3 V5 V7) (V1 V2 V5 V6) (V2 V4 V6 V8) (V3 V4 V7 V8))
                 (V1 V3 V5 V7) (V1 V2 V5 V6) (V2 V4 V6 V8) (V3 V4 V7 V8)
                  COLOR: RED    COLOR: RED     COLOR: RED    COLOR: RED
PAINT (BLK (TOP BOTTOM) BLUE)
      0    1         1    0

    (V1 V2 V3 V4 V5 V6 V7 V8)
      HAVE: ((SIDES COLOR RED) AND ((TOP BOTTOM) COLOR BLUE))
                                     I-SPACE: ((V1 V2 V3 V4) (V5 V6 V7 V8))
                                               (V1 V2 V3 V4) (V5 V6 V7 V8)
CUT-INTO (BLK 16 ONE-INCH CUBES)                COLOR: BLUE   COLOR: BLUE
                                0
    (V1 V2 V3 V4 V5 V6 V7 V8)
      SLICES: (L-R-VERTICAL B-F-VERTICAL)
                 APVAL: 3       APVAL: 3
      PARTS: ((CUBES THREE-D (1 1 1)) HOW-MANY 16)
* QUESTIONS TO PROBLEM NO. 1 *
ENCODE-QUESTION ()
                 10
    OBJ0
      HAVE:  ((FACES COLOR RED) AND (FACES COLOR BLUE))
      TYPE:  CUBES
      HOW-MANY:  0
```

0.8	
1.4	B3: Both red and blue faces.
1.8	
3.0	B4: Ah, the sides are red and the top and bottom are blue.
1.5	
1.0	B5: Cut them that way.
0.8	
1.2	B6: Well it would be all of them on the border.
2.5	
0.4	B7: Uh
3.4	
0.6	B8: So that's
0.5	
0.7	B9: How many on the border?
0.9	
1.0	B10: One, on the periphery?
0.6	
1.7	B11: One, two, three, four,
1.0	
0.9	B12: five, six, seven,
0.6	
1.1	B13: eight, nine, ten,
1.2	
0.6	B14: eleven, twelve.

```
MATCH-SCHEMA ()
             00
     S-OBJ21  ((SIDES COLOR RED) AND ((TOP BOTTOM) COLOR BLUE))
     I-OBJ21  (((V1 V3 V5 V7) (V1 V2 V5 V6) (V2 V4 V6 V8) (V3 V4 V7 V8))
                   AND ((V1 V2 V3 V4) (V5 V6 V7 V8)))
     ACTIVE-ELEMENTS:  ((V5 V6 V7 V8) (V1 V2 V3 V4) (V3 V4 V7 V8)
                        (V2 V4 V6 V8) (V1 V2 V5 V6) (V1 V3 V5 V7))
     ACTIVE-ATTRIBS:  (2 ((COLOR BLUE) (COLOR RED)))
CSET (FOCI TOP)
      0       J
pB (BLK)
   0   0
```





```
TAL (P[illegible]-PIECES)
     1         n
```

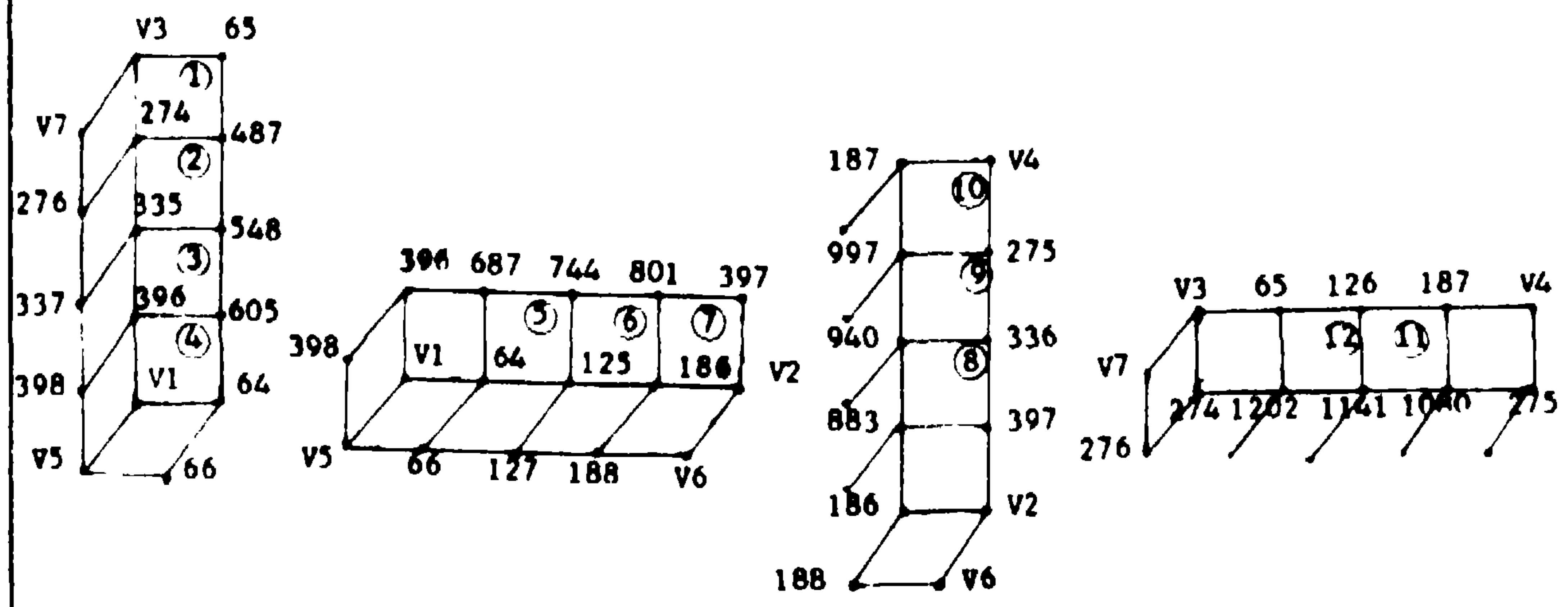


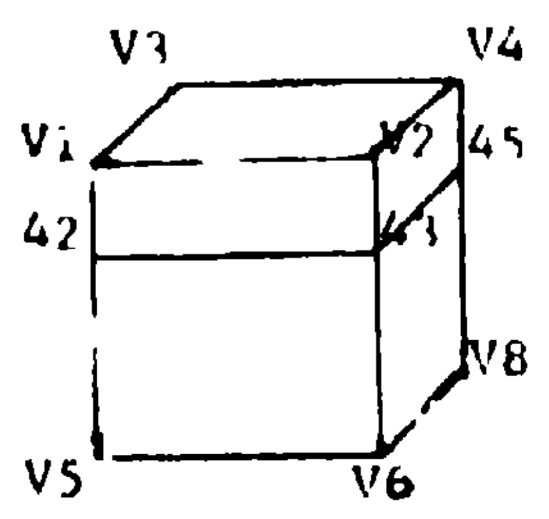
FIGURE 3A Printout of the first level structure of BORDER-PIECES (BVT #1)

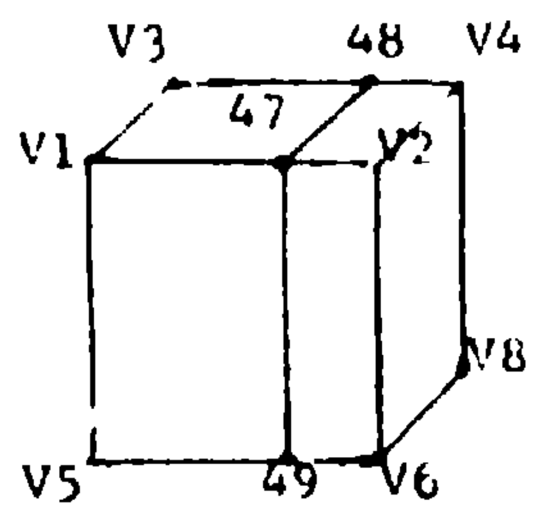
```
THE BORDER-PIECES ((V1 V3 V5 V7 G00064 G00065 G00066 G00067) (G00396 G00397 G00398 G00399 V1 V2 V5
V6) (G00186 G00187 G00188 G00189 V2 V4 V6 V8) (V3 V4 V7 V8 G00274 G00275 G00276 G00277))

HAVE

(V1 V3 V5 V7 G00064 G00065 G00066 G00067)
  SLICES:  (B-F-VERTICAL)
  DEPTH:  4
  WIDTH:  1
  HEIGHT:  1
  THREE-D:  (1 1 4)
  ACTIVE:  (2 ((V1 V3 G00064 G00065) (V1 V3 V5 V7)))
  LEFTSDE:  (V1 V3 V5 V7)
  RIGHTSDE:  (G00064 G00065 G00066 G00067)
  TOP:  (V1 V3 G00064 G00065)
  BOTTOM:  (V5 V7 G00066 G00067)
  FRONT:  (V1 G00064 V5 G00066)
  BACK:  (V3 G00065 V7 G00067)
  FOCUS:  TOP

WHERE THE TOP IS COLOR BLUE AND WHERE THE LEFTSDE IS COLOR RED.

(G00396 G00397 G00398 G00399 V1 V2 V5 V6)
  SLICES:  (L-R-VERTICAL)
  DEPTH:  1
  WIDTH:  4
  HEIGHT:  1
  THREE-D:  (1 4 1)
  ACTIVE:  (2 ((G00396 G00397 V1 V2) (V1 V2 V5 V6)))
  BACK:  (G00396 G00397 G00398 G00399)
  FRONT:  (V1 V2 V5 V6)
  TOP:  (G00396 G00397 V1 V2)
  BOTTOM:  (G00398 G00399 V5 V6)
  LEFTSDE:  (G00396 V1 G00398 V5)
  RIGHTSDE:  (G00397 V2 G00399 V6)
  FOCUS:  TOP

WHERE THE TOP IS COLOR BLUE AND WHERE THE FRONT IS COLOR RED.

(G00186 G00187 G00188 G00189 V2 V4 V6 V8)
  SLICES:  (F-B-VERTICAL)
  DEPTH:  4
  WIDTH:  1
  HEIGHT:  1
  THREE-D:  (1 1 4)
  ACTIVE:  (2 ((G00186 G00187 V2 V4) (V2 V4 V6 V8)))
  LEFTSDE:  (G00186 G00187 G00188 G00189)
  RIGHTSDE:  (V2 V4 V6 V8)
  TOP:  (G00186 G00187 V2 V4)
  BOTTOM:  (G00188 G00189 V6 V8)
  FRONT:  (G00186 V2 G00188 V6)
  BACK:  (G00187 V4 G00189 V8)
  FOCUS:  TOP

WHERE THE TOP IS COLOR BLUE AND WHERE THE RIGHTSDE IS COLOR RED.

(V3 V4 V7 V8 G00274 G00275 G00276 G00277)
  SLICES:  (R-L-VERTICAL)
  DEPTH:  1
  WIDTH:  4
  HEIGHT:  1
  THREE-D:  (1 4 1)
  ACTIVE:  (2 ((V3 V4 G00274 G00275) (V3 V4 V7 V8)))
  BACK:  (V3 V4 V7 V8)
  FRONT:  (G00274 G00275 G00276 G00277)
  TOP:  (V3 V4 G00274 G00275)
  BOTTOM:  (V7 V8 G00276 G00277)
  LEFTSDE:  (V3 G00274 V7 G00276)
  RIGHTSDE:  (V4 G00275 V8 G00277)
  FOCUS:  TOP

WHERE THE TOP IS COLOR BLUE AND WHERE THE BACK IS COLOR RED.
```

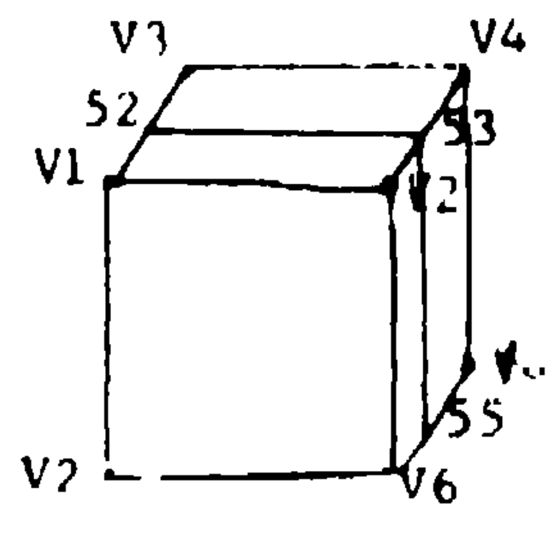
BVT #21 "Three slices are made in a 3 inch cube. One is made horizontally one inch from the top, another vertically one inch from one side, and a third one inch from the front,
4. How many piecee are there?
5. How many 1 inch cubes are there⁹
6. How many 1 inch by 2 inch by 2 inch pieces are there?
7. How many 2 inch cubes are there?

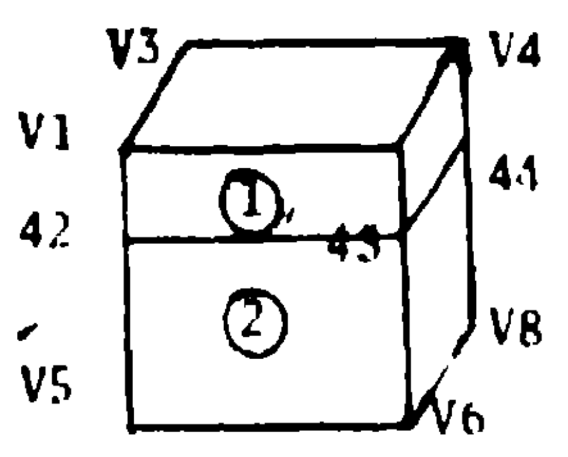
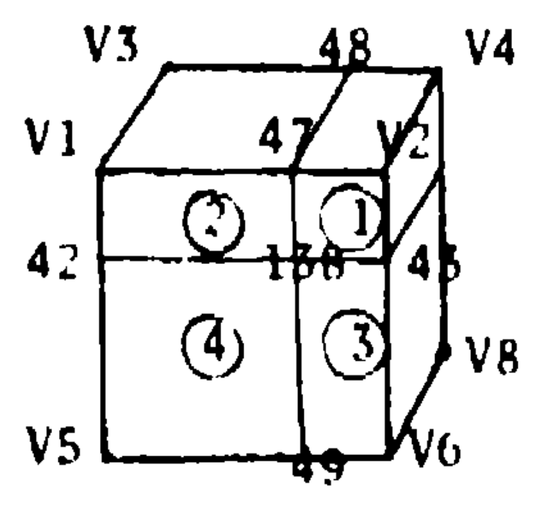
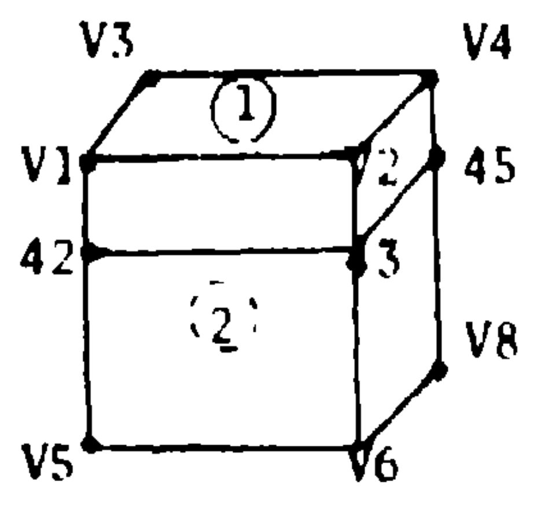
Secs.	Protocol
2.7	E1: "All right, three slices are made in a 3 inch cube.
1.4	
1.3	E2: A 3 inch cube
0.6	
1.8	E3: 3 by 3 by 3 I presume that means.
1.5	
1.5	E4: One is made horizontally
1.1	
4.6	E5: 1 inch fron the top, all right, E6: another vertically one inch from one side, all right,
0.8	
1.7	E7: and a third vertically 1 inch from the front.
1.1	
2.3	E8: All right, I think, if I can keep them all.
1.4	
1.7	F1: 4. How many pieces are there? Uhh.

```
IMAGE (S-BLOCK)
      0      0
IMAGE (I-BLOCK)
      4      0


DIMENSIONALIZE (BLK 3 3 3 )
               0       0
   (V1 V2 V3 V4 V5 V6 V7 V8)        (V1 V2 V3 V4) ETC.
      DEPTH: 3                        DEPTH:  3
      WIDTH: 3                        WIDTH:  3
      HEIGHT: 3                       TWO-D:  (3 3)
      THREE-D:  (3 3 3)               NAME:  TOP
CUT-UP (BLK HORIZONTAL (1 INCH FROM TOP))
   (V1 V2 V3 V4 V5 V6 V7 V8)
      SLICES:  (HORIZONTAL)
                 I-SPACE:((G00042 G00043 G00044 G00045))
                 APVAL: 1
CUT-UP (BLK R-L-VERTICAL (1 INCH FROM RIGHTSDE))
       SLICES:  (HORIZONTAL R-L-VERTICAL)
       ((G00047 G00048 G00049 G00050))
CUT-UP (BLK F-B-VERTICAL (1 INCH FROM FRONT))
       0                 1                    10


       SLICES:  (HORIZONTAL R-L-VERTICAL F-B-VERTICAL)
       ((G00052 G00053 G00054 G00055))
+ QUESTIONS TO PROBLEM NO. 2 +
ENCODE-QUESTION ()
                00
       OBJ0
         TYPE:  PIECES
         HOW-MANY:  0
```

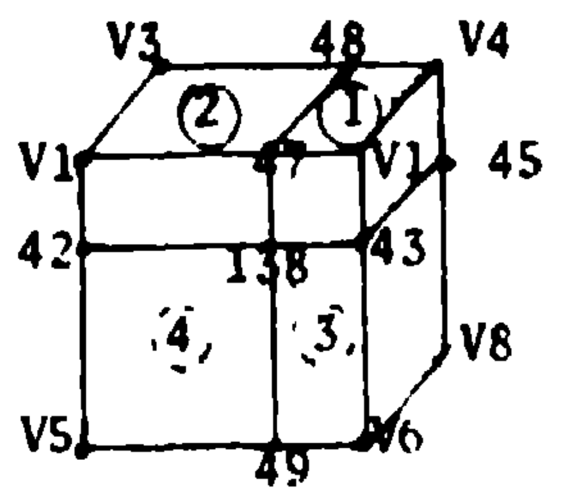


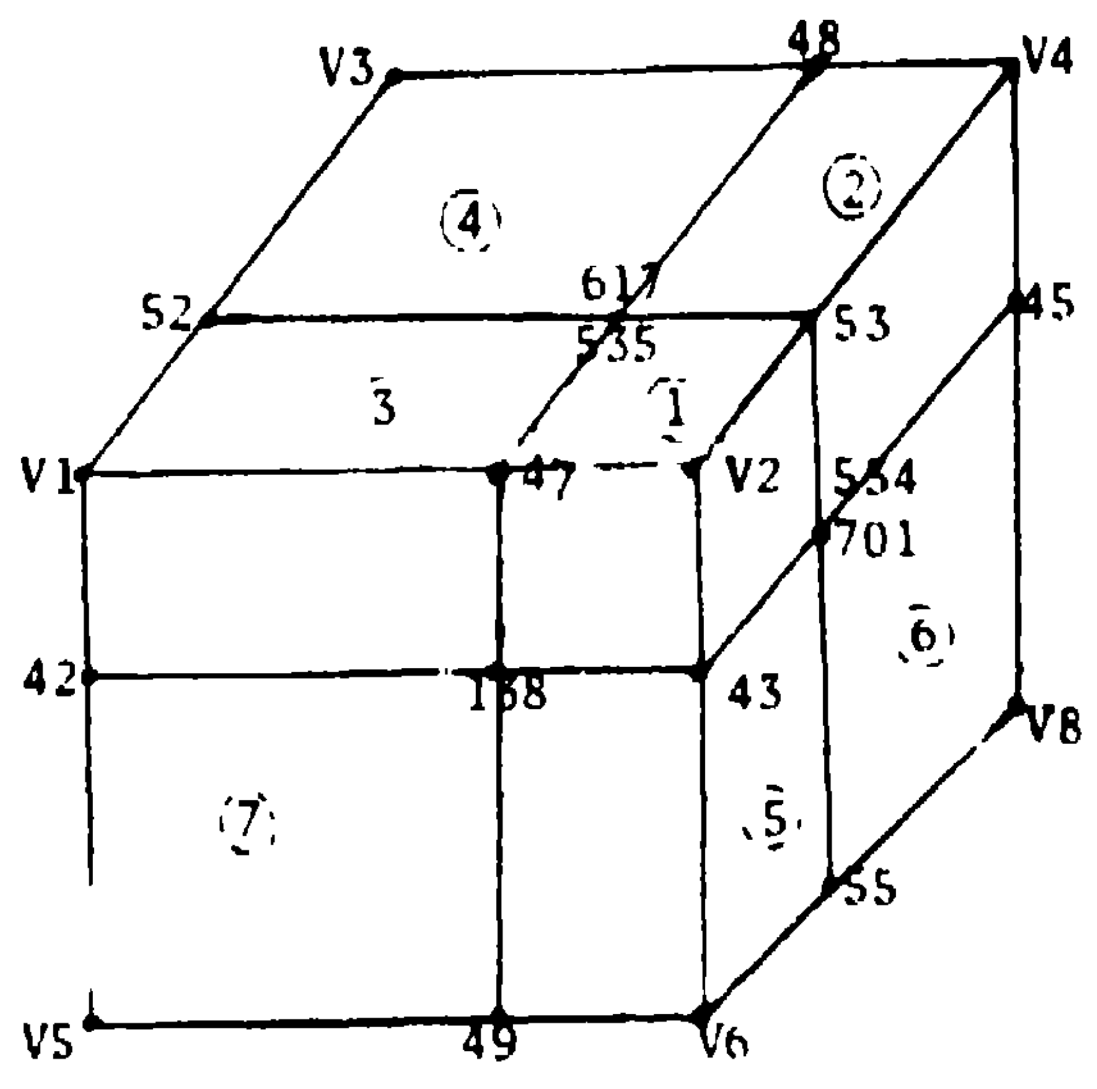




5.5	F2: One is made horizontally 1 inch from the top; that makes two pieces; F3: another vertically one inch from the side,	MATCH-SCHEMA () 00 CSET (FOCI FRONT) 0 0 TAL (BLK) 0 0  V3 V4 V1 42 44 1 43 2 V8 V5 V6
1.5		
0.8	F4: that makes:	 V3 48 V4 V1 47 V2 2 1 42 138 43 4 3 V8 V5 49 V6
3.0		
2.5	F5: one, two, three, four pieces.	
1.4		
2.0	F6: And a third one inch from the front.	
2.3		
0.5	F7: Uhm.	
5.1		
2.5	F8: Oh gosh I find that very hard.	CUTS = (DISEMBODIED) THE FOCUS FACE TOP IS DISEMBODIED.
2.7		
1.6	F9: All right,.let's see if I can work it out.	CSET (FOCI TOP) 0 0 TAL (BLK) 0 0  V3 V4 1 V1 V2 45 42 43 2 V8 V5 V6
2.7		
4.2	F10: We have one horizontal cut which cuts it into two pieces; F11: if you take this one down	
3.1		
0.5	F12: and	
3.9		
1.6	F13: then you have a piece there and a piece there	
1.2		

1.2	F14: and the pieces underneath,
0.6	
0.5	F15: right.
1.1	
1.1	F16: I have that visualized.
0.5	
1.6	F17: Now you take a pi-, cut down the front
3.7	
1.1	F18: and that just doubles it.
1.6	
5.8	F19: But we had four before and essentially we're cutting them all in, not in half, but in thirds, so I presume eight.
1.1	
1.6	F20: We have that piece and that piece,
0.8	
3.4	F21: that piece and that piece, F22: and then that reciprocally behind: eight.
2.7	





```
OBJO
  HOW-MANY: 8
  TYPE: PIECES
```

```
PIECES
  APVAL, ((V2 G0043 G00047 G00138 G00053 G00534 G00535 G00536) (G00053 G00534 G00535 G00536 V4
G00045 G00048 G00139) (G00047 G00138 V1 G00042 G00617 G00618 G00052 G00619) (G00617 G00618 G00053
G00619 G00048 G00139 V3 G00044) (G00043 V6 G00138 G00049 G00701 G00055 G00702 G00703) (G00701
G00055 G00702 G00703 G00045 V8 G00139 G00050) (G00138 G00049 G00042 V5 G00782 G00783 G00784 G00054)
(G00782 G00783 G00784 G00054 G00139 G00050 G00044 V7))
```

```
(V1 V2 V3 V4 V5 V6 V7 V8)
   PIECES:   ((V2 600043 G00047 600138 G00053 G00534 G00535 G00536) (600053 G00534 G00535 G00536 V4
G00045 G00048 G00139) (G00047 G00138 v1 G00042 G00317 G00618 G00052 G00619) (G00617 G00618 G00052
G00619 G00048 G00139 V3 G00044) (G00043 V6 G00138 G00049 G00701 G00055 G00702 G00703) (G00701
G00055 G00702 G00703 G00045 V8 G00139 G00050) (G00138 G00049 G00042 V5 G00782 G00783 G00784 G00054)
(G00782 G00783 G00784 G00054 G00139 G00050 G00044 V7))
   SOTPIECE:   (G00042 G00043 G00044 G00045 V5 V6 V7 VB)
   TOPPIECE:  (V1  V2 V3 V4 G00042 G00043 G00044 G00045)
   SLICES:    (HORIZONTAL  R-L-VERT1CAL  F-B-VERTICAL>
   DEPTH:   3
   WIDTS:  3
   HEIGHT  :  3
   THREE-D:    (3  3  3)
   TOP:    (VI  V2  V3  V4)
   80TTOM:    (V5  V6  V7  VB)
   LEFTSOE:  (V1   V3  V5  V7)
   FRONT:  (V1   V2  V5  V6)
   RIGHTSOE:    (V2   V4   V6   V8)
   BACK:   (V3    V4   V7   v8)
   NAME  :  8LK
   PIECEOI  ((V1   V2  V3  V4  V5  V6  V7  V8))

 OBJO
    HOWMANY  :  8
    TYPE'   PIECES
```