DOCUMENT RESUME

ED 152 329 IR 005 883

AUTHOR Gomez, Fernando; Chandrasekaran, B.
TITLE Closeness and Reformulation: Two Cognitively-Based Ideas for Problem-Solving.
INSTITUTION Ohio State Univ., Columbus. Computer and Information Science Research Center.
SPONS AGENCY Air Force Office of Scientific Research, Washington, D.C.
REPORT NO OSU-CISRC-TR-11
PUB DATE Jul 77
GRANT AFOSR-72-2351
NOTE 25p.

EDRS PRICE MF-$0.83 HC-$1.67 Plus Postage.
DESCRIPTORS *Artificial Intelligence; *Cognitive Processes; Decision Making; Learning Theories; *Problem Solving; Research; Thought Processes

ABSTRACT

The approach to problem solving which is presented has two components--closeness and reformulation. The closeness measure is a cognitively-based heuristic function. It incorporates the notion of what cognition notices as the structural difference between two situations. The problem-solver attempts to close the structural gap, and once this is done in the right way, other things fall into place. The reformulation measure provides the problem-solver with new ways of looking at the goal or task, and is mediated by the closeness measure. These two ideas are applied to several traditional problems. (Author/DAG)

DOCUMENT RESUME

ED 152 329 IR 005 883

AUTHOR Gomez, Fernando; Chandrasekaran, B.
TITLE Closeness and Reformulation: Two Cognitively-Based Ideas for Problem-Solving.
INSTITUTION Ohio State Univ., Columbus. Computer and Information Science Research Center.
SPONS AGENCY Air Force Office of Scientific Research, Washington, D.C.
REPORT NO OSU-CISRC-TR-11
PUB DATE Jul 77
GRANT AFOSR-72-2351
NOTE 25p.

EDRS PRICE MF-$0.83 HC-$1.67 Plus Postage.
DESCRIPTORS *Artificial Intelligence; *Cognitive Processes; Decision Making; Learning Theories; *Problem Solving; Research; Thought Processes

ABSTRACT

The approach to problem solving which is presented has two components--closeness and reformulation. The closeness measure is a cognitively-based heuristic function. It incorporates the notion of what cognition notices as the structural difference between two situations. The problem-solver attempts to close the structural gap, and once this is done in the right way, other things fall into place. The reformulation measure provides the problem-solver with new ways of looking at the goal or task, and is mediated by the closeness measure. These two ideas are applied to several traditional problems. (Author/DAG)

(OSU-CISRC-TR-11)

CLOSENESS AND REFORMULATION: TWO COGNITIVELY-BASED IDEAS FOR PROBLEM-SOLVING

by

Fernando Gomez
and
B. Chandrasekaran

B. Chandrasekaran's research
supported by Air Force Office of Scientific Research
Grant 72-2351

Computer and Information Science Research Center
The Ohio State University
Columbus, Ohio 43210

July, 1977

TABLE OF CONTENTS

ABSTRACT

The approach that we present to problem-solving has two components: closeness and reformulation. The closeness measure is a cognitively-based heuristic function, and reformulation provides the problem-solver with new ways of looking at the goal and is mediated by the closeness measure. We apply the proposed ideas to many problems that have traditionally been used to test problem-solving ideas.

> Generally speaking, there is first a situation
> S_1, the situation in which the actual thought process starts, and then, after a number of steps,
> S_2, in which the process ends, the problem is solved.
>
> Let us consider the nature of situation 1 and situation 2 by comparing them, and let us then consider what goes on between, how and why. Clearly the process is a transition, a change from S_1 into S_2. S_1, as compared with S_2, is structurally incomplete, involves a gap or a structural trouble, whereas S_2 is in these respects structurally better; the gap is filled adequately, the structural trouble has disappeared; it is sensibly complete as against S_1.
>
> When the problem is realized, S_1 contains structural strains and stresses that are resolved in S_2. The thesis is that the very character of the steps, of the operations, of the changes between S_1 and S_2 springs from the nature of the vectors set up in these structural troubles in the direction of helping the situation, of straightening it out structurally. This is quite in contrast to processes in which some steps, some operations coming from various sources and going in various directions, may lead to the solution in a fortuitous, zigzag way.

-- Max Wertheimer, <u>Productive Thinking</u>

PREFACE

The Computer and Information Science Research Center of the Ohio State University is an interdisciplinary research organization consisting of staff, graduate students, and faculty of many University departments and laboratories. This report describes research undertaken in cooperation with the Department of Computer and Information Science.

The research of B. Chandrasekaran was supported by AFOSR Grant 72-2351.

5

INTRODUCTION

In this paper we present a new approach to problem-solving and apply it to a variety of problems that have traditionally been a test-bed for problem-solving ideas. The reader who is impressed by the role cognition plays in human problem-solving will notice that the new approach has a great deal of cognitive relevance. On the other hand, the person who is performance-minded will have opportunities to see the scope and generality of the proposed ideas through the examples, which include the Father and Sons Task, various water-jug problems, blocks problems with interaction of goals and the Elementary Algebra Task.

A SEARCH PROCEDURE

For convenience of exposition, we confine ourselves to problems representable in state space and use the terminology of Nilsson [1]. The problem-solver would require as primitive only a heuristic function which would enable it to compute a "closeness measure" between two states, denoted $C(s_1,s_2)$ for two states s_1 and s_2. It could be a numerical measure but it need not be so. The closeness measure gives the problem-solver the ability to decide which of a set of candidate nodes to expand next. The formation of the closeness criterion corresponds to an "understanding" of the problem and thus would be task-dependent. Actually, we shall see that for classes of tasks, essentially the same closeness measure would be applicable.

The search is controlled by a modified depth-first algorithm. In the following, we will not explicitly state the activities normal to this class of algorithms, such as the establishment of back pointers. The initial state is s_o, the goal state is G, PARENT(s) is the parent node of s. For each node s that is expanded, BACKUP(s) is a set of nodes to back up to if expansion of node s results in no expandable successors, i.e., a cut-off has occurred.

1) $s \leftarrow s_o$, BACKUP(s) is empty set.

2) Expand s, delete from the successors nodes already generated, and let

the remaining successors be the set S. If S is empty, go to 5). Else, check to see if G∈S. If yes, exit with success. Else go to 3).

3) Choose the node in S that is closest to G (decide ties arbitrarily) and set the value of s to this node.

4) BACKUP(s) ← S - {s}. If BACKUP(s) is non-empty, go to 2). Else set BACKUP(s) ← BACKUP(PARENT(s)) and go to 2).

5) (S empty means a cut-off has occurred.) If BACKUP(s) is empty, exit with failure. Otherwise set S ← BACKUP(s) and go to 3).

The basic idea of the procedure is simple. As a node is expanded, the set of successors is pruned to eliminate any elements identical to nodes already generated. If the goal is not found among the remainder, the successor closest to the goal is chosen for expansion next. If a node has no expandable successors (i.e., if the pruned set of successors is empty), a cut-off has occurred, and the closest node from its unexpanded siblings is chosen. If this is not possible, trace back through the ancestors until one with a set of unexpanded siblings is found, and choose the closest from it for further expansion. If no such node is found, exit with failure. This kind of back-up incorporates the notion of "pursuing a line of thought".

We have given a rather simple search procedure for purposes of exposition. One of the desirable modifications would be when the closeness measures for two candidate states are equal. In the above procedure, we have broken the ties arbitrarily, and the search would go down the chosen node. However, a more intelligent procedure would be to expand both of them one level down, see if any further insight through closeness of successors can be obtained before a committment is made to a "line of thought".

The interpretation of the closeness measure is what distinguishes it from other heuristic functions used to order nodes, such as the cost function, h, of Nilsson. Our closeness measure is related to what is cognitively considered the "essential" part of the description.

SYMBOLIC CLOSENESS

There is a simple type of closeness which we call symbolic closeness and which is the appropriate closeness measure for a class of problems, which includes the Father and Sons Task, the Logic Task, the Missionaries and Cannibals Task and the Elementary Algebra Task. In all these cases, certain symbols are abstracted from the state description as having been deemed "essential". The closeness measure is obtained by matching lists of these abstracted symbols. For instance, let $L = \{A,B,C\}$, $L_1:\{A,B,C,D\}$, $L_2:\{A,B\}$, and $L_3:\{A,B,E\}$ be such lists of abstracted symbols for states s, s_1, s_2, and s_3; then $C(s,s_1) = C(s,s_2) < C(s,s_3)$; i.e., s_1 and s_2 are equally close to s, requiring change in one symbol to achieve symbolic closeness, but s_3 is farther from s, needing changes in two symbols. Let us illustrate by means of the Father and Sons Task [2].

Example 1. - Father and Sons Task

The problem is: "A father weighing 200 pounds and two sons each weighing 100 pounds wish to cross a river. The only conveyance available is a boat of capacity 200 pounds. Father and sons can operate the boats individually." Our system starts with the following representation: Initial State is RIGHT(F,S1,S2,Boat), LEFT(None); goal is LEFT(F,S1,S2,Boat). The list used for closeness is the LEFT list of candidate state, i.e., this list is matched with that of the goal state. The more matches, the closer. The tree appears as Figure 1.

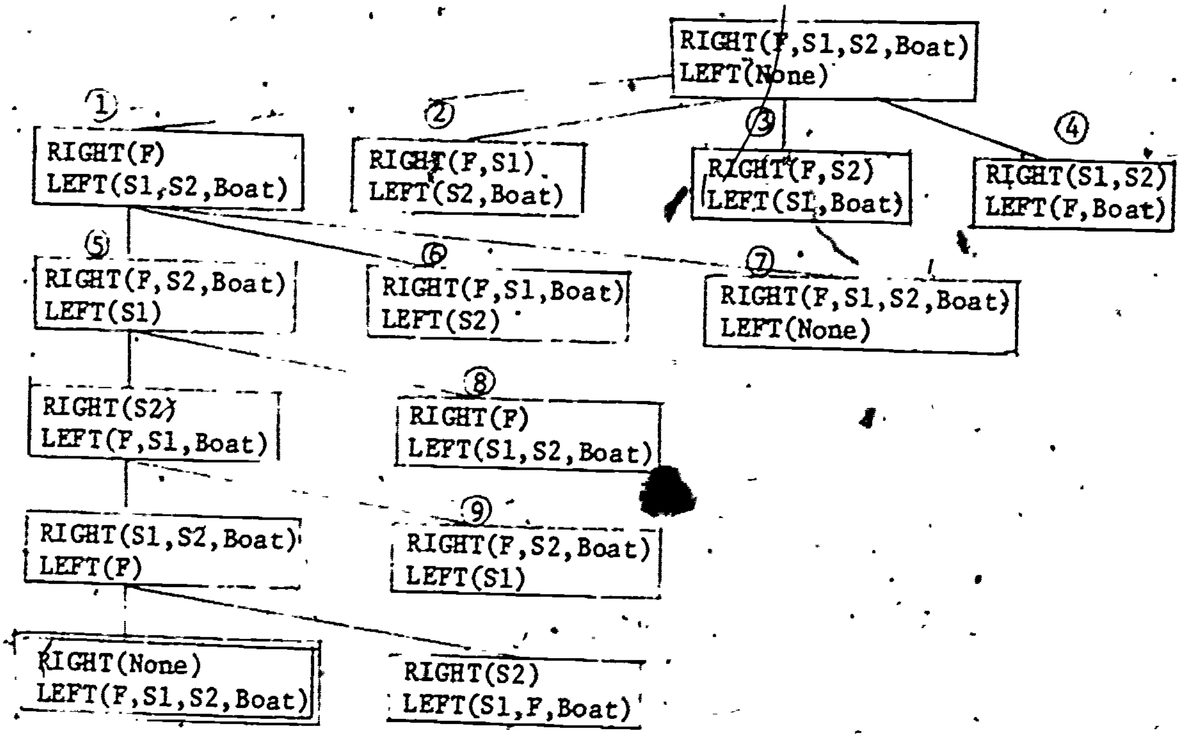

RIGHT(F,S1,S2,Boat)
LEFT(None)
1
RIGHT(F)
LEFT(S1,S2,Boat)
2
RIGHT(F,S1)
LEFT(S2,Boat)
3
RIGHT(F,S2)
LEFT(S1,Boat)
4
RIGHT(S1,S2)
LEFT(F,Boat)
5
RIGHT(F,S2,Boat)
LEFT(S1)
6
RIGHT(F,S1,Boat)
LEFT(S2)
7
RIGHT(F,S1,S2,Boat)
LEFT(None)
RIGHT(S2)
LEFT(F,S1,Boat)
8
RIGHT(F)
LEFT(S1,S2,Boat)
RIGHT(S1,S2,Boat)
LEFT(F)
9
RIGHT(F,S2,Boat)
LEFT(S1)
RIGHT(None)
LEFT(F,S1,S2,Boat)
RIGHT(S2)
LEFT(S1,F,Boat)

Figure 1

In this problem, the only time closeness is used is to choose node 1 over nodes 2, 3, and 4. Nodes 6, 7, 8, and 9 were generated earlier and lose out to their siblings. This performance is striking in comparison with that of GPS on the same problem [2].

Symbolic closeness is applicable to problems where certain tokens need to be present or absent in the state representation. Later we shall see that symbolic closeness works very successfully in the Algebra Task. It is our working hypothesis that a taxonomy of problems exists based on the closeness measures that they call for, and the process of "understanding" the problem corresponds to "framing" the problem in a cognitive map within the appropriate taxonomic unit, this resulting in the relevant closeness heuristic.

In this first example, the cognitive role was restricted to noting the presence or absence of symbol tokens. In the next, quantity is introduced in the closeness measure.

QUANTITATIVE CLOSENESS

Example 2. - A Water Jug Task [2]

"Given a five-gallon jug and an eight-gallon jug, how can precisely two gallons be put into the five-gallon jug? Since there is a sink nearby, a jug can be filled from the tap and can be emptied by pouring its contents down the drain. Water can be poured from one jug into another, but no measuring devices are available other than the jugs themselves."

The representation is - initial state {J5(0), J8(0)}, goal J5(2). In this case, the essential element is not simply the presence or absence of tokens, but the concept of quantity. Let J5(x) and J5(y) be the components of states s and s_1 where s_1 is a successor of s. If x is less than 2 (since the goal description is J5(2) in this example) the successors of s will be judged on the basis of their contribution in increasing the contents of J5. Then, s_1 scores a match, in closeness calculation, if $y > x$. If x is greater than 2, the above argument will be reversed. The alternatives for expansion will be scored in terms of this measure of closeness. Notice the role played by the parent of node s_1 in calculating the closeness of s_1. It is another aspect of the notion of "pursuing a line of thought".

The tree that is generated is given in Figure 2.

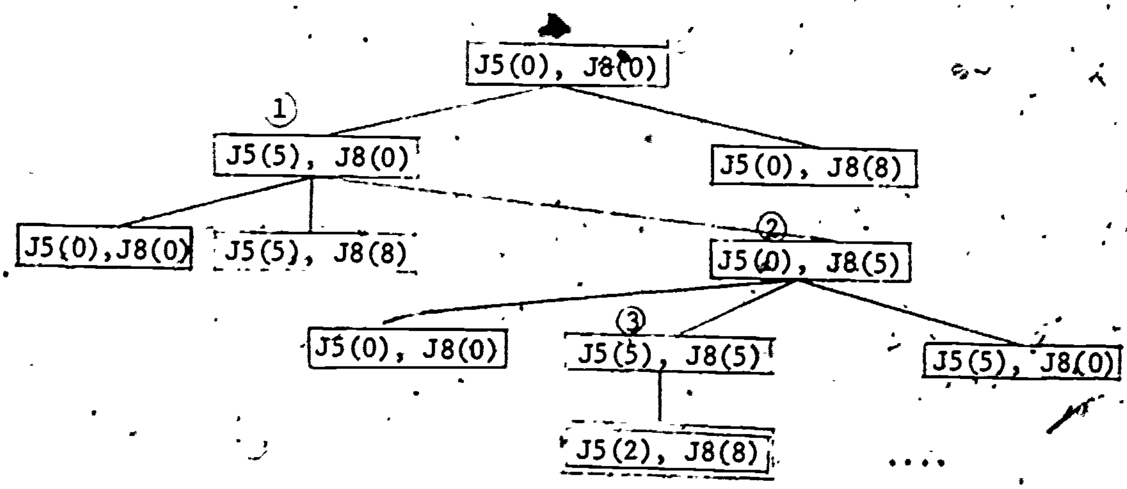


Figure 2

The circled numbers near the nodes represent the order of expansion. Ernst and Newell [2] comment on the behavior of GPS in solving this problem: "The use of differences in this task seems to be a rather ineffective means of guiding the problem-solving ... GPS might need some additional problem-solving mechanism, e.g., planning, in order to be more proficient at the task." The reader might notice that to the extent that planning involves abstracting the "essential" from the "inessential" in problem-solving, our system has the rudiments of planning in the concept of closeness.

ORDER-BASED CLOSENESS

In the literature recently, the problem of interaction of goals has received some attention. Sacerdoti [3], Sussman [4], Tate [5], and Warren [6] consider as a prototypical example the following problem in the BLOCKS world.

Example 3.

```
                          A
Change   C          to    B
         A    B           C
         ---------TABLE   ---------TABLE
```

Only one operator is available: PUTON(X,Y), for which X should have cleartop. It is generally pointed out that GPS fares poorly in this problem and others which are characterized by the so-called interaction of goals. This term arises since GPS tends to view the goal as consisting of a conjunction of two subgoals: (ON B C) and on ON(A B), but if one subgoal is achieved, it will have to be undone to achieve the other subgoal. In our opinion, this problem arises in this case because of a weakness in traditional representations of the problem, which do not permit imparting of some essential information to the system which is intuitively available to the human. We shall solve the problem using two different notions of closeness, one we feel might correspond to that of a person who has lived all his life in a world of no gravity and other capturing our intuition of the role played by the order of blocks.

Let the representation be:

initial state		goal state	
	ON C A		ON A B
	ON A TABLE		ON B C
	ON B TABLE		ON C TABLE

The first notion of closeness is simply: the more ON statements the state description shares with the goal state the closer. The tree in Figure 3 is generated.

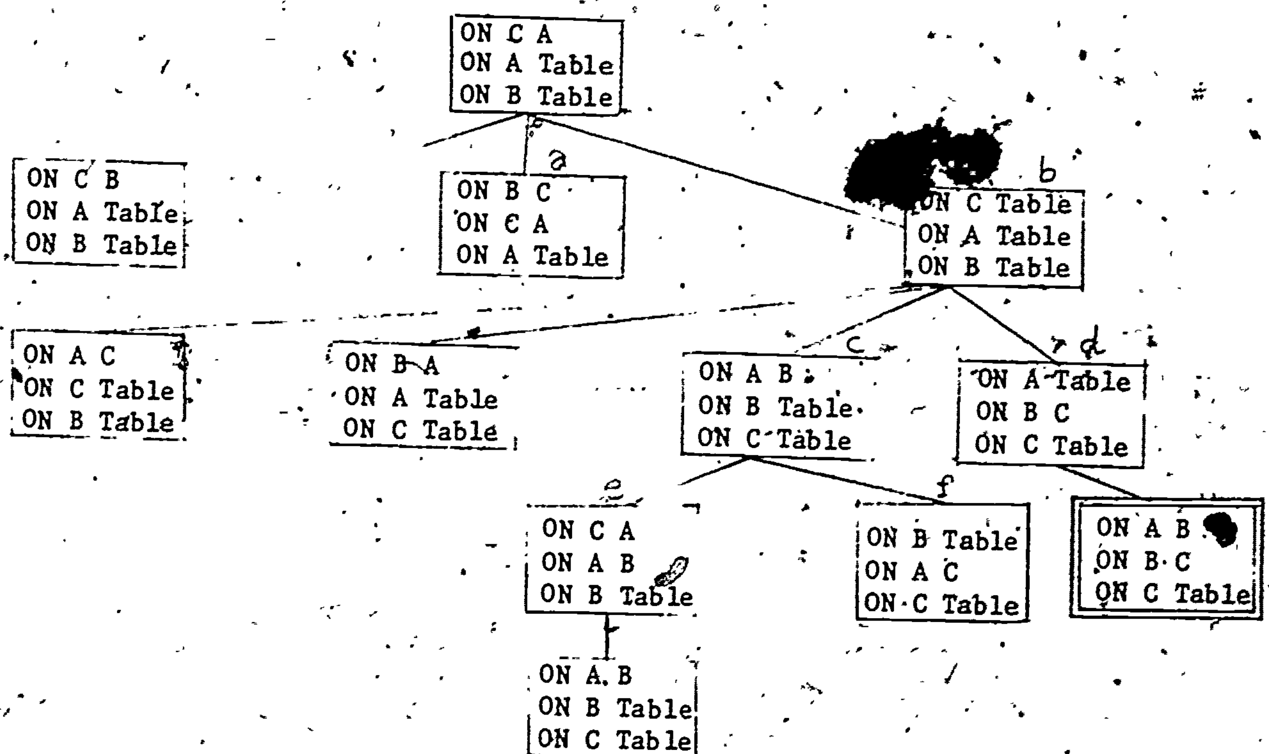
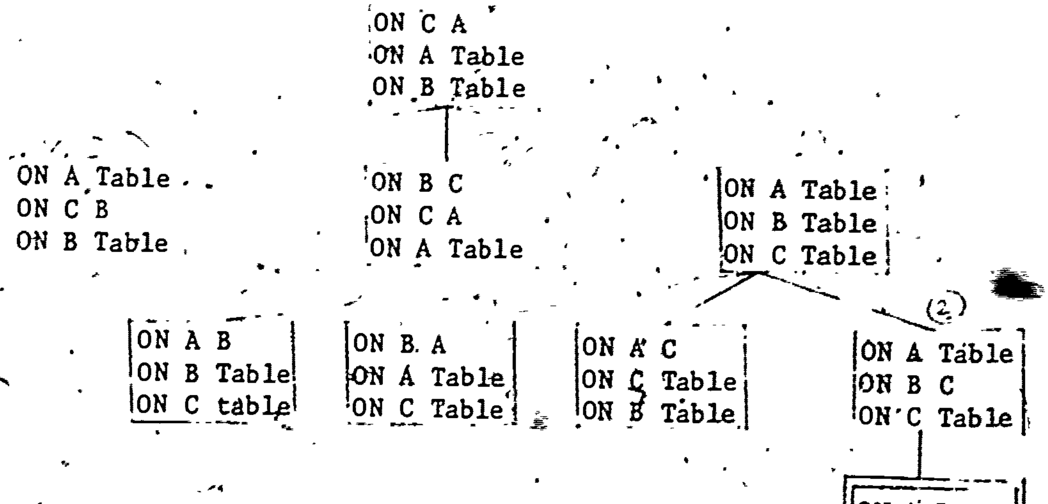


Figure 3

Some explanations: Expansion of node a is not shown: it simply results in the initial state. Nodes c and d are equally close. We have assumed the worst and let c be selected, resulting in successors e and f. Node f is recognized as having been generated, expansion of e results in a cut-off. The system backs up to d which leads straight to the goal.

Now let us consider a more sophisticated notion of closeness for this class of problems. The reason why humans solve this problem with remarkable ease is that they do not regard the three components of our goal descriptions as equivalent and independent. There is a piece of knowledge that they bring into the situation, something that is a product of having lived in a world with gravity, and that can be incorprated into the closeness measure. The new notion of closeness is: If ON(X_n,Table) is part of the goal description, then a state having ON(X_n,Table) is closer than another state which does not have ON(X_n, Table), whatever the matchings of other component descriptions. Similarly, a state having {(ON(X_{n-1},X_n), ON(X_n,Table)} is closer than another state which does not have {(ON(X_{n-1},X_n), ON(X_n,Table)}, whatever the matchings of other components; and so on. The new tree is given in Figure 4. A significant reduction in search over Figure 3 is seen.

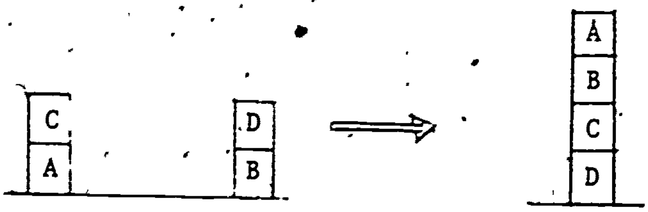


(Nodes which are the same as those generated earlier are not shown)

Figure 4

Example 4

Let us consider the following more complicated case. For simplicity we switch to a pictorial representation of states without loss of generality. The problem is taken from [3] and is as follows.



The tree which results after search with the closeness measure incorporating order is given in Figure 5.

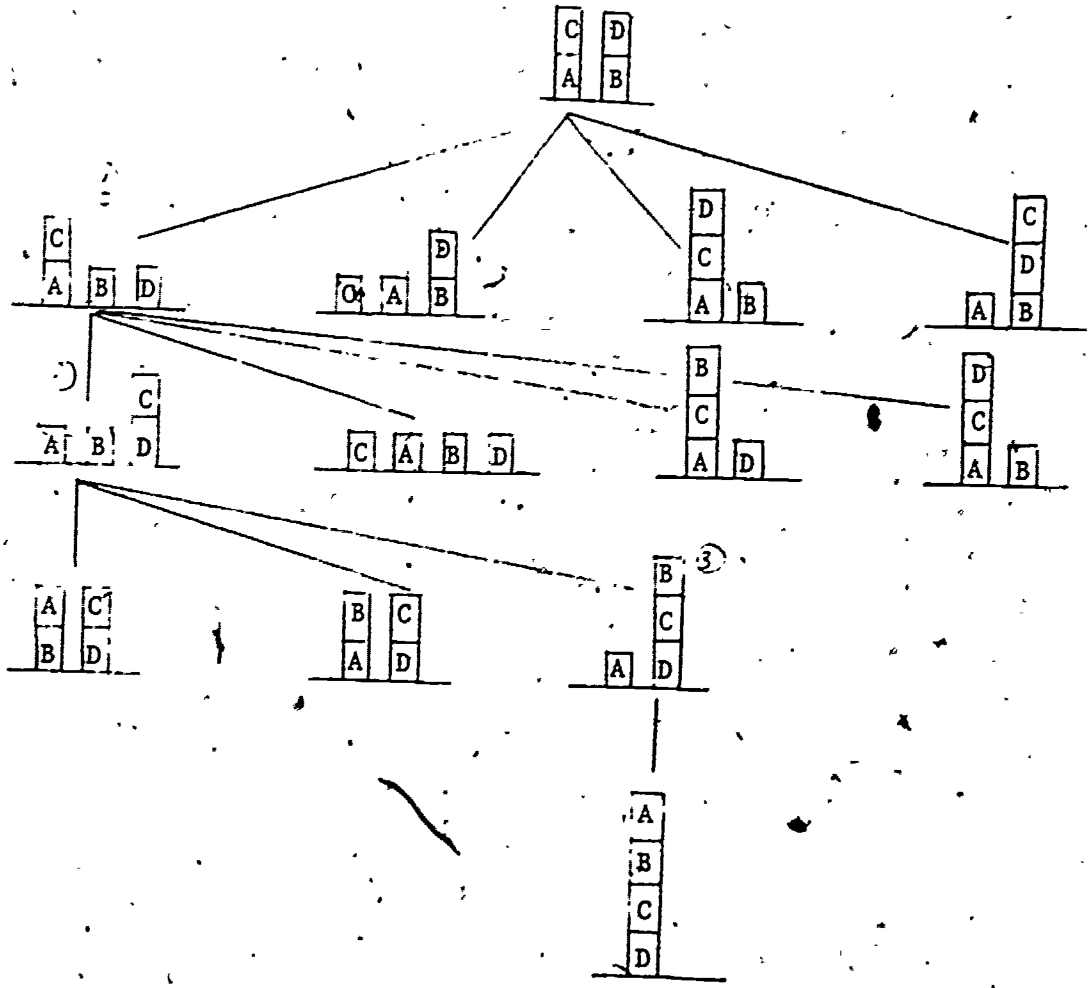


Figure 5

The solution is remarkably straightforward. Node $\hat{1}$ is closer than its siblings and thus chosen, because 1 satisfies (ON D, Table). Similarly node $2'$ is chosen over its siblings because it satisfies (ON C D) and (ON D Table) and so on.

As Sacerdoti [3] points out, goal interaction can be avoided by planning. As mentioned earlier, and as can be seen from the examples in this section, our notion of closeness gives the system the rudiments of planning.

REFORMULATION

The human problem solver, while engaged in a search with whatever concepts of closeness provided him by his own cognitive system, is also considering possible reformulations of the problem. This reformulation guides the search in such a way that both the original problem and the reformulations are simultaneously kept in mind. Our concept of closeness promises to provide a smooth way of integrating these modes of problem-solving.

Again, let us confine ourselves to problems whose solution process can be completely modeled as a state-space search (like all the examples so far.) A classical notion here is "working back from the goal". In fact, the reformulations we talk about will be, for this class of problems, states which lead to the goal state. However, in order to cut down the search space backwards, one needs some sort of criterion. Since the only primitive notion available to our problem-solver is that of closeness, the reformulation in general, and working back from the goal in particular, need to be anchored to that measure. In the Logic Task, reformulation based on closeness is more than working back from the goal and in fact leads naturally to problem-reduction. For now, however, we see reformulation as a way of gaining insight by backing up from the goal, examining what results, and using it to guide forward search. Wertheimer [7] says, in discussing the performance of young Gauss, at age six, summing 1 + 2 + ... 10 in a new way, "In the process the various items clearly gain a new meaning; they appear functionally determined in a new way. Nine is no longer viewed as 8 plus 1, it has become 10 minus 1, and so on." Reformulation is an attempt to give the system the ability to see the goal in a new way.

Our criterion for a meaningful reformulation of a goal G as G' is that: i) there exist permissible operators taking G to G' and vice versa, and ii) G and G' differ in some <u>essential</u> respects, or, in the language of closeness, $C(S,G) \gtrless C(S,G')$ where S is, say, the initial state, and $C(x,y)$ is the measure of closeness between states x and y. $C(S,G) = C(S,G')$ implies that in the respects deemed essential by the cognitive system, G and G' are not meaningfully different, and thus looking at G' will not yield any insight.

The suggestion that G' qualifies as a reformulation even if S is less close from G' than from G might be puzzling at first. However, creativity consists in looking at possibilities which are against convention or are counter-intuitive. Reformulation is a way of "shaking up" the goal representation for possible insights and a more "difficult" goal qualifies as a reformulation for this reason.

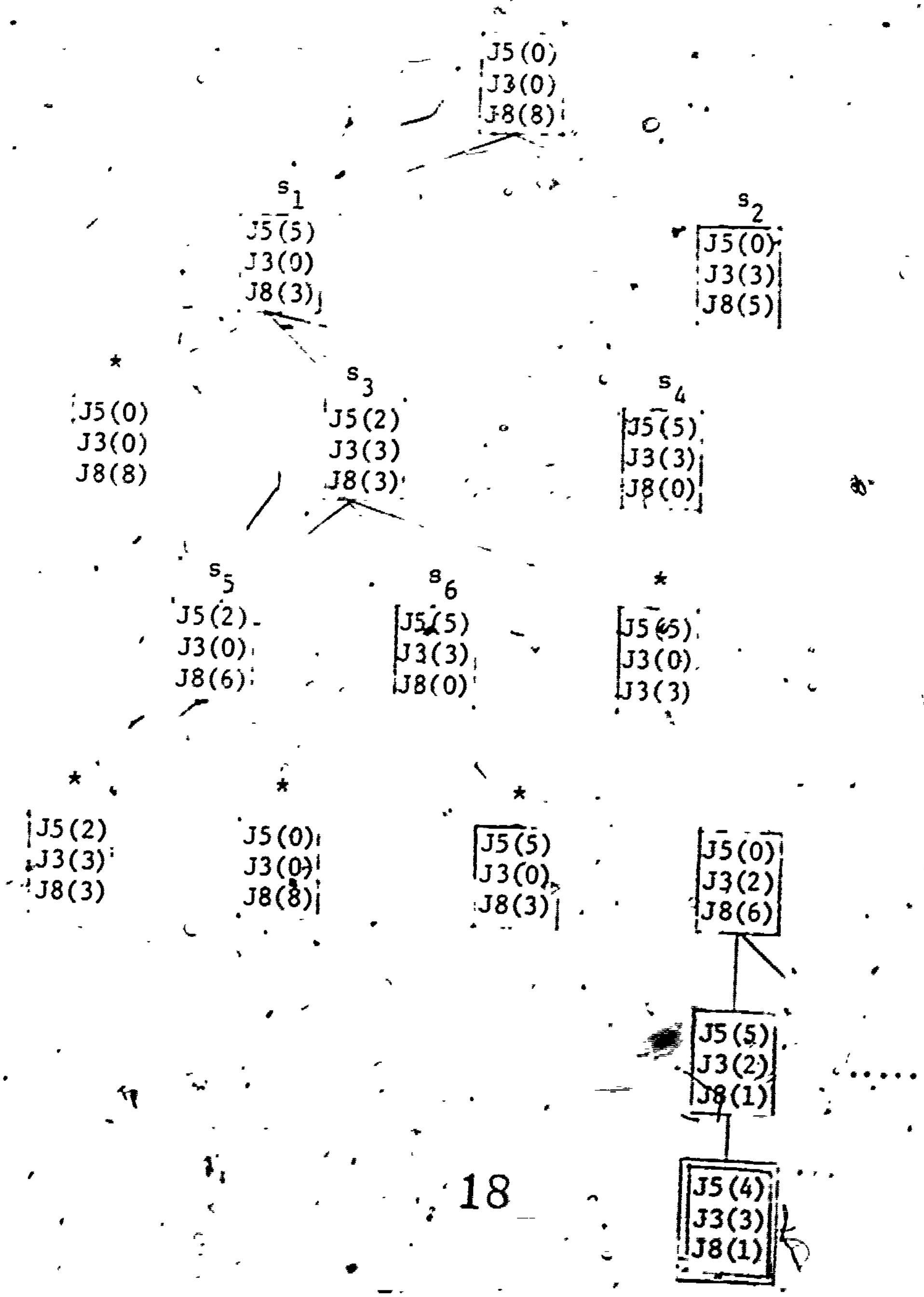
Suppose a set of meaningful reformulations $\{G_1,\ldots,G_n\}$ have been generated. Before reformulation, closeness for a state s would be $C(s,G)$. However, after the reformulation, the modified closeness measure would be $\min\{C(s,G), C(s,G_1), \ldots, C(s,G_n)\}$. For each candidate node, this modified closeness would be computed, and the node with the smallest measure would be selected. Perhaps the ideas will become clearer after the next example.

<u>Example 5</u>

Let us take the following "ale-jug" problem [2]. The initial state is {J5(0), J3(0), J8(8)}, and the goal is {J5(4), J3(0), J8(4)}. However, jugs can neither be filled with ale from any tap nor can they be emptied by pouring ale down the drain (heaven forbid!).

The basic closeness measure is similar to that used in Example 2, but now we have two goal components. The extension is best illustrated by considering the two successors of the initial state, s_1:{J5(5), J3(0), J8(3)} and s_2:{J5(0), J3(3), J8(5)}, and the goal state. Considering J5, we wish to increase its contents, thus s_1 will score a match. Similarly considering J8, the goal state requires reducing its contents. Here both s_1 and s_2 will score matches. Now considering both of them together, s_1 will be deemed closer than s_2.

The following are two candidates for reformulation: G_1 = {J5(4), J3(3), J8(1)}, and G_2 = {J5(1), J3(3), J8(4)}, and both staisfy the conditions for reformulation. The forward search based a reformulation now proceeds and the tree is generated as in Figure 6.



Choice between s_1 and s_2 in favor s_1 would be done with or without reformulation, with the basic notion of closeness available to the system. Let us consider the choice between s_3 and s_4 [nodes marked * are nodes previously generated and closed]. Without reformulation, both s_3 and s_4 will score equally in the closeness calculation. This is because between s and goal, the contents of J5 and J8 need to decrease, s_3 will score one match with respect to J5 and s_4 also one, but with respect to J8. On the other hand, with reformulation, the closeness measure can be illustrated as follows.

		G	G_1	G_2
s_3	J5	x	x	x
	J8			
s_4	J5			
	J8	x		

That is, s_3 is now deemed closer, and expanded. State s_6 is recognized as identical to s_4 and is not expanded. If pursuing s_5 would have led to a cut-off (in this case it doesn't), then s_4 would be pursued. As it is, however, the goal is reached quickly.

We have applied reformulation to a variety of other problems with success. In particular, the Father and Sons Task and the Missionaries Task quickly result in solutions, when successive reformulation, combined with symbolic closeness is employed. There is hardly any search. The Elementary Algebra Task [8] is handled elegantly by reformulation. We now proceed to that task.

ELEMENTARY ALGEBRA TASK

The rewrite rules, as they appear in [8], are given below:

R1. A+B := B+A

R2. A+(B+C) := (A+B)+C

R3. (A+B)-B := A

R4. A := (A+B)-B

R5. (A-B)+C := (A+C)-B

R6. (A+B)-C := (A-C)+B

The closeness measure used is the symbolic closeness, i.e., a list is formed of the token symbols appearing in the expression and used to compute the closeness measure as explained earlier. The operators do not appear in the list, i.e., they are abstracted out.

Given a theorem to prove of the form LEFT STRING := RIGHT STRING, only rules R3 and R4 are potentially applicable for reformulation of the goal, since they are the only ones that change symbolic closeness. In R4, B plays the role of a variable standing for any expression. Each reformulation is at first kept in the form containing the variable, say Y.

The first task of the executive* is to achieve symbolic closeness. It expands the initial node (i.e., the LEFT STRING). If R4 is used, there will be variables. A determination is made of the possible substutions for the variables, say X, such that the resulting expression is as close as possible to the goal. These possible substitutions are remembered, but the successor is kept in the form containing X.

Now a substitution for the variable Y in a given reformulation is made so as to result in an expression as close as possible to the successors of the initial node. Since these successors contain the generic variable X, the substitution for Y will in general be in a form containing X. This is done for each reformulation and the set of reformulations is now used to guide forward search so as to achieve grouping closeness. Once grouping closeness is achieved, X is substituted for by the possible substitutions that were determined earlier, to see if the two expressions which are symbolically and grouping-wise close, are in fact identical. If so, the problem is solved. Otherwise, other branches of the search tree are pursued.

Let us illustrate it with an example that had the longest solution time for the Quinlan-Hunt system.

*The search procedure described earlier needs to be given additional capabilities for this class problems, such as substitution for free variables. The needed changes should become clear as the executive is described.

Example 7 Prove $(A-C)-(B-C) := A-B$

The Quinlan-Hunt system had proved five theorems up to this point. We shall assume that our system has access to the same theorems, in particular Theorem 5, which is $(A-B)+C := A+(C-B)$.

The first stage of the reformulation of the goal is as follows.

$$A-B \xrightarrow{R4} \begin{cases} ((A+Y)-Y)-B \\ A-((B+Y)-Y) \\ ((A-B)+Y)-Y \end{cases}$$

The initial state is $(A-C)-(B-C)$. The only rule applicable is R4, but it can be applied in seven different ways :

1. $(((A+X)-X)-C)-(B-C)$
2. $(A-((C+X)-X))-(B-C)$
3. $(((A-C)+X)-X)-(B-C)$
4. $(A-C)-(((B+X)-X)-C)$
5. $(A-C)-(B-((C+X)-X))$
6. $(A-C)-(((B-C)+X)-X)$
7. $(((A-C)-(B-C))+X)-X$

Of the possible substitutions for X, substitution of A or B would be the only ones consistent with maximizing closeness to the goal. This is noted and put away for future use.

The variable Y in the reformulations is now to be substituted for in such a way as to maximize closeness to one or more of the seven successors. It can be seen that substitution of $(X \pm C)$ would result in symbolic closeness being achieved between the successors and the reformulations. All the seven successors are now equally close to the reformulations, which are, after substitution, the following :

I. $((A+(X\pm C))-(X\pm C))-B$

II. $A-((B+(X\pm C))-(X\pm C))$

III. $((A-B)+(X\pm C))-(X\pm C)$

Figure 7 shows the results of forward search, assuming that the first of the successors is selected. The last expression in the tree is identical

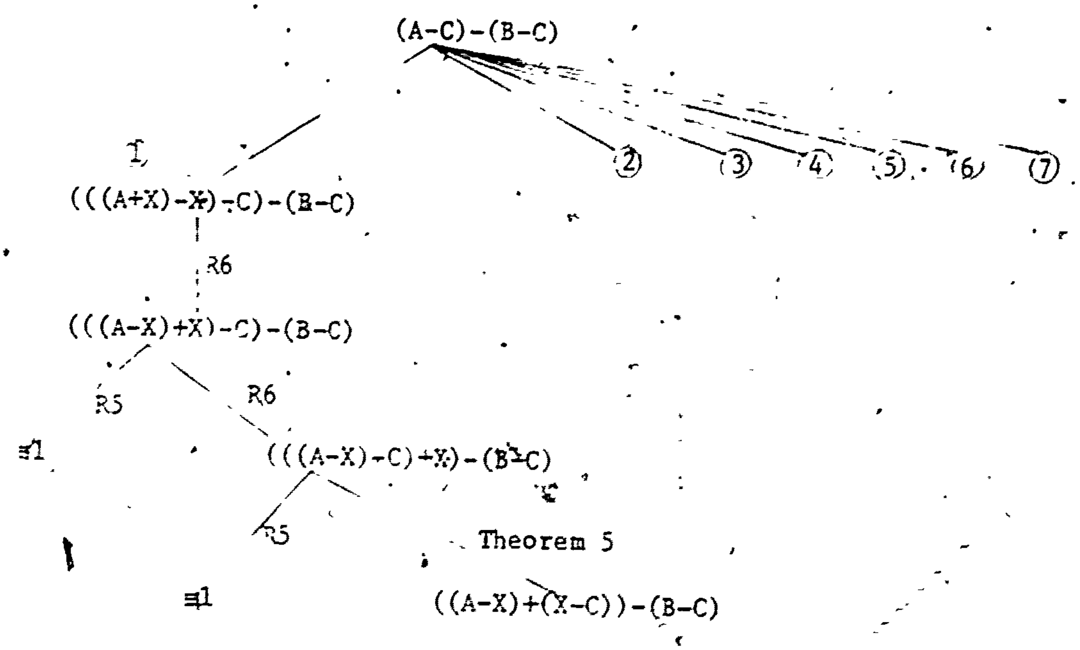


Figure 7

to reformulation III, with the substitution X=B and the choice of the - sign in the substitution Y=X±C. The theorem is proved.

Choice of any of the successors 2 to 6 would result in a cut-off. Figure 8 shows this for 2.

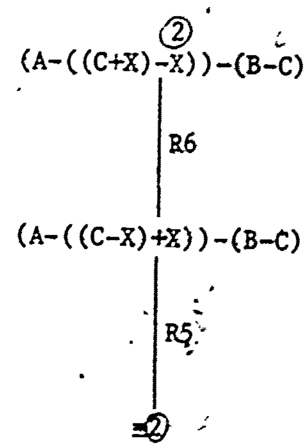


Figure 8

On the other hand, if 7 had been chosen, there would be success again. Figure 9 shows this. The expression pointed to by the arrow in the figure matches reformulation I for X=B and Y=B-C.

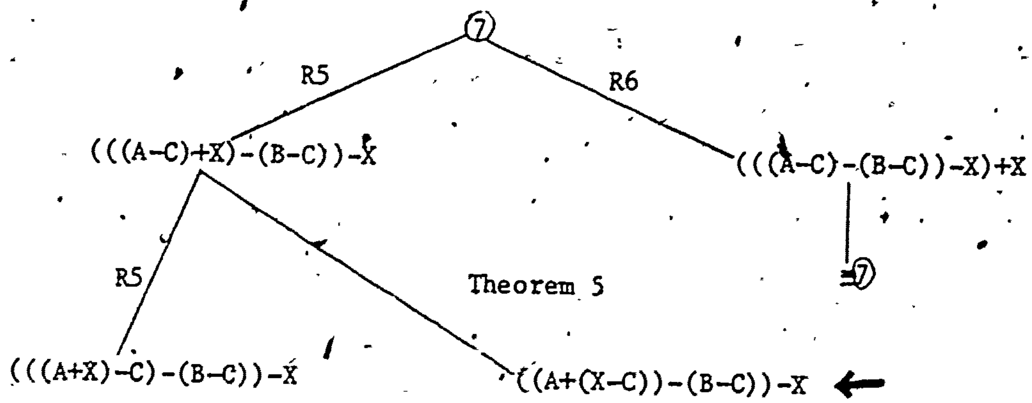


Figure 9

We gave the problem-solver only the notion of symbolic closeness. If we had given, in addition, a notion of grouping closeness (people use that in making choices for this class of problems), 1 or 7 would have been expanded first. However, it is not clear to the authors at present how a general concept of grouping closeness can be formulated. It is a subject of current investigation. It would be easy to come up with a grouping closeness measure that will work for this case, but that would have had an <u>ad hoc</u> flavor.

CONCLUDING REMARKS

This paper is based on the view that at the very base of any problem-solving activity there is a cognitive component. The various problem-solving modes such as search, planning and problem-reduction are not independent, disjointed activities, but work in a coherent way, mediated by input from cognition. A task of any problem-solving theory is to uncover this cognitive role, which tends to be hidden under the accumulation of a number of high-level heuristics.

The closeness concept incorporates our notion of what cognition notices as the structural difference between two situations. The problem-solver attempts to close the structural gap, and once this is done in the right way, other things fall into place. Reformulation is a powerful way of obtaining a different view of the task.

There are several aspects of these ideas which need further investigation. It is not clear how to formalize the specific conditions under which reformulation is to be activated in a problem-solver. To some extent, an absolute committment is not wise, since a problem-solving theory should capture not only what is common to intelligent problem-solving, but also should provide for individual differences, and individual difference plays a role in the invocation of reformulation. Nevertheless, more understanding of this aspect is needed.

Means-ends analysis is a useful component of problem-solving, though we do not assign it a dominant role. We are currently studying how this component can be smoothly integrated into our system.

The taxonomy of closeness itself is a matter of great interest. Other kinds of closeness measures will be clearly needed and a systematic investigation should provide a great deal of insight into the structure of the cognitive base of problem-solving.

REFERENCES

[1] Nilsson, N.J., Problem-Solving Methods in Artificial Intelligence, Mc-Graw Hill, 1971.

[2] Ernst, G.W., Newell, A., GPS : A Case Study in Generality and Problem-Solving, Academic Press, 1969.

[3] Sacerdoti, E., "The nonlinear nature of plans," Advance Papers of IV

International Joint Conference on Artificial Intelligence, 1975, pp.206-214.

[4] Sussman, G.J., A Computer Model of Skill Acquisition, American Elsevier, 1975.

[5] Tate, A., "Interacting goals and their use," Advance papers of IV International Joint Conference on Artificial Intelligence, 1975, pp. 215-218.

[6] Warren, D.H.D., "WARPLAN : a system for generating plans," memo no. 76, Department of Computational Logic, University of Edinburgh, June 1974.

[7] Wertheimer, M., Productive Thinking, Harper & Brothers, 1945, p.91.

[8] Quinlan, J.R., Hunt, E.B., "A formal deductive problem-solving system," JACM 15, 4, 1968, pp. 625-646.