# CARPS, A PROGRAM WHICH SOLVES CALCULUS WORD PROBLEMS

by

EUGENE CHARNIAK

A.B., University of Chicago
(1967)

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

July, 1968

Signature of Author Eugene Charniak
Department of Electrical Engineering, July 12, 1968

Certified by Joel Moses
Thesis Supervisor

Accepted by
Chairman, Departmental Committee on Graduate Students

# ACKNOWLEDGMENTS

The work reported herein was supported in part by Project MAC, an MIT research program sponsored by the Advanced Research Projects Agency, Department of Defense, under Office of Naval Research, contract Number Nonr-4102(01), and in part by the National Science Foundation. Reproduction in whole or in part is permitted for any purpose of the United States Government.

I would like to thank Professor Moses, who originally suggested the project, for his supervision of this thesis and the use of his programs. My appreciation also goes to Adolfo Guzman, both for the use of CONVERT and his explanations of its workings.

## TABLE OF CONTENTS

# CARPS, A PROGRAM WHICH SOLVES CALCULUS WORD PROBLEMS

by

EUGENE CHARNIAK

Submitted to the Department of Electrical Engineering on July 12, 1968
in Partial fulfillment of the requirements for the Degree of Master of Science.

## ABSTRACT

A program was written to solve calculus word problems. The program, CARPS (CAlculus Rate Problem Solver), is restricted to rate problems. The overall plan of the program is similar to Bobrow's STUDENT, the primary difference being the introduction of "structures" as the internal model in CARPS. Structures are stored internally as trees. Each structure is designed to hold the information gathered about one object.

A description of CARPS is given by working through two problems, one in great detail. Also included is a critical analysis of STUDENT.

Thesis Supervisor: Joel Moses

Title: Assistant Professor of Electrical Engineering

## I. INTRODUCTION

The problems connected with computer understanding of general natural language input are at present beyond our abilities. Yet by limiting the context of the English input to a specific topic the problem is simplified in many ways. In a limited context a given word will have fewer possible meanings, so that there is less difficulty in deciding between meanings of a word. We also have the advantage of needing only a limited subset of English vocabulary and grammar. Finally, within a limited context certain words can be used to give clues to the meaning of a sentence as well as to the manner in which a sentence should be broken up into smaller sentences.

Bobrow's STUDENT (2), a program which assumed that the input was an algebra word problem, was able to solve a large variety of problems with a remarkably small program. We have to a large extent taken our ideas from Bobrow's work. An understanding of STUDENT is sufficiently important to our work that we shall spend the second chapter analyzing Bobrow's program.

The research described in this paper had as its goal the creation of a program which solves freshman calculus word problems. The program, CARPS (Calculus Rate Problem Solver), is restricted to rate problems. CARPS is written in two languages. The bulk of the coding is in LISP. There are, however, large sections which require a great deal of pattern matching, something for which LISP is not particularly powerful. These sections were written in CONVERT (3,4), a language especially designed for pattern matching. Because CONVERT is imbedded in LISP it was an especially convenient choice since we could easily switch back and forth between the two languages. Both of these languages were available on the Project MAC PDP-6 time sharing system which was used in this research. This system, which has a quarter million words of core storage, gave us a decided advantage over Bobrow, whose program had to fit into a 32K 7094 LISP system, whereas ours wallows in the comparative luxury of 45K of memory.

Though CARPS is a fairly complex program, its basic organization is relatively straightforward. To demonstrate the underlying principles let us show how it would solve the following particularly simple problem. The parentheses and slashes are required by the PDP-6 LISP system.

```
(A SHIP IS 30.0 MILES SOUTH OF POINT O AND TRAVELING WEST
AT 25.0 MILES PER HOUR /. HOW FAST IS THE DISTANCE FROM THE
SHIP TO O INCREASING?)
```

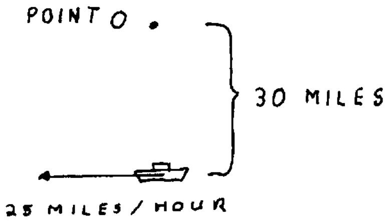


Fig. 1

The diagram of this problem is for the benefit of the reader. CARPS does not use diagrams.

The program is divided into five sections. The primary goal of the first section is to tag words with their part of speech. At the same time the program accomplishes several other tasks such as checking for words indicating the type of problem it has been handed. Just before it turns the transformed problem over to the next section it will print out

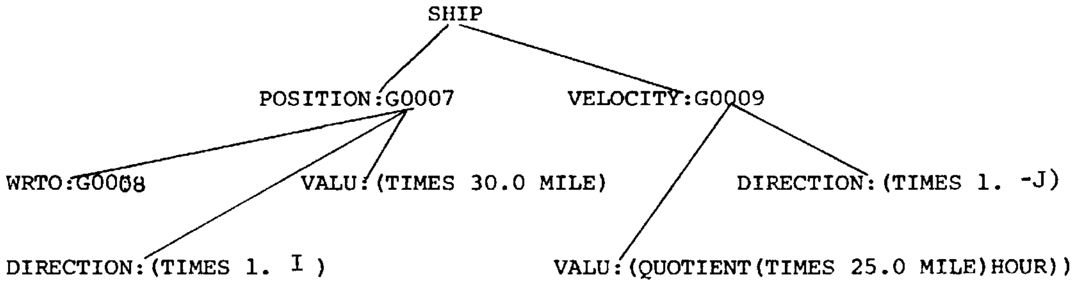
```
(THE PROBLEM WITH TAGS ON IS)
(((A SHIP (IS VERB) 30.0 (MILE UNIT) (SOUTH PNOUN)POINT 0 AND
(TRAVELING VERB) (WEST PNOUN) (AT PREP) 25.0 (MILE UNIT) PER
(HOUR UNIT)) (1.))
(((HOW QWORD) (FAST RWORD) (IS VERB) THE DISTANCE (FROM PREP)
THE SHIP TO 0 (INCREASING VERB)) (2.)))

(THE PROBLEM TYPE IS)
DISTANCE
```

At the moment the program is familiar with two types of problems, DISTANCE and VOLUME. The second section of the program takes the output of the first section and breaks the sentences into simple sentences. After it has done so it will print;

```
(THE SIMPLIFIED SENTENCES ARE)
(((A SHIP (IS VERB) 30.0 (MILE UNIT) (SOUTH PNOUN) POINT 0)(1.))
((A SHIP (TRAVELING VERB) (WEST PNOUN)) (1.))
((A SHIP (TRAVELING VERB) (AT PREP) 25.0 (MILE UNIT) PER
(HOUR UNIT)) (1.))
(((HOW QWORD) (FAST RWORD) (IS VERB) THE DISTANCE (FROM PREP)
THE SHIP TO 0 (INCREASING VERB)) (2.)))
```

Note that the first sentence has been broken into three, while the second has been left unchanged.

The third section is responsible for taking these simple sentences and transforming them into the model of the problem which the computer must have to solve it. The model used in the program is composed of equations and "structures". A "structure" is basically a tree which has as its head the name of some object, and at various levels beneath the head all the information the program was able to abstract from the problem. (The second level corresponds to the property list of the head atom. The third level corresponds to the property lists of the atoms on the second level.) In our problem there is only one structure. It is most easily visualized in the following form.



The expressions G0007, G0008, etc. are symbols generated by the LISP system. They are commonly called GENSYMS, and are used as fillers in the structures.

Looking at the right hand node of the structure we see that the velocity of the ship is 25 miles per hour, and this velocity is in the -J direction. (The program assumes the co-ordinate system

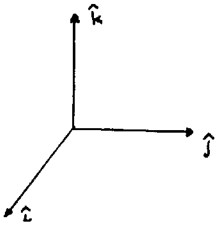


Fig. 2

so -J would be west as it should be.)

The left hand node can be intrepreted in the same way, except that WRTO stands for "With Respect To". This means that the ship's position was measured with respect to POINT 0 (G0008).

The fourth section will generate the equations which the final section will solve. In this problem it must change the expression (DISTANCE (SHIP) (G0008)) to an actual distance equation. After it has done so the program prints out

```
(THE EQUATION SET IS)
((EQUAL (G0012) (EXPT (PLUS (TIMES 899.99998 (EXPT MILE 2.))
(TIMES 624.99998 (EXPT T1M 2.) (EXPT MILE 2.)
(EXPT HOUR -2.))) 0.5)))
```

Noting that T1M is the symbol for time we see that this is just the equation written in more usual notation is

$$X = \sqrt{900\ \text{MILES}^{2} + 625\ T^{2}\ \text{MILES}^{2}/\ \text{HOUR}^{2}}$$

The final section of the program will differentiate, simplify, and finally print out

```
(THE ANSWER IS)
0.
```

This, of course, with either a little thought, or a little algebra can be shown to be the correct answer. CARPS took 41 seconds of machine time to solve this problem.

As we mentioned previously the major influence to our work has been Bobrow's STUDENT. There is however other work which deserves mention. One may look at CARPS as a problem in utilizing a limited context to enable the program to accept natural language. In this respect Weizenbaum's work on ELIZA (9) is relevant. ELIZA is a program which accepts scripts which contain directions for analyzing the input sentences and producing a response. One ELIZA script was designed to act as a Rogerian psychiatrist. A user of the program with this script would sit down and type statements to the machine. The program would first analyze these statements for key words. With each key word are associated transformations, and the context of the key word which would make this transformation applicable. For example, if you typed YOU ARE VERY HELPFUL it would notice the key word YOU and eventually transform the sentence to the response WHAT MAKES YOU

THINK THAT I AM VERY HELPFUL. If there are no key words in the sentence the program generates a content free remark, or goes back to a key word found previously and produces some output based on it.

As Weizenbaum has pointed out ELIZA accomplishes its goal by hiding its lack of knowledge of the "real world". A second ELIZA script (10) was written to do exactly the opposite. A typical conversation with it is

```
The area of a circle is pi times the radius of the circle
squared.
The area of a sphere is pi times the diameter of the sphere
squared.
The diameter of an object is twice the radius of the object.
The radius of a ball is 10.
What is the area of the ball.
THE ASSUMPTION THAT BALL IS CIRCLE IS CONSISTENT-SHALL I GO
ON? no
THE ASSUMPTION THAT BALL IS SPHERE AND OBJECT IS CONSISTENT-
SHALL I GO ON? yes
IT'S 1256.6359
```

Once again the analysis is essentially a transformation based upon key words.

The second ELIZA might also be considered a natural language question answering system. In this respect it is somewhat like SIR, (Semantic Information Retrieval) (7), a program written by B. Raphael. A conversation with SIR looks remarkably similar to the above with ELIZA.

```
A nose is part of a person
I UNDERSTAND
A nostril is part of a nose
I UNDERSTAND
A professor is a teacher
I UNDERSTAND
A teacher is a person
I UNDERSTAND
Is a nostril part of a professor Q
YES
```

Contrary to ELIZA the input formats in SIR are strictly defined. As such SIR is really a study in storage of semantic information rather than work which concentrates on the recognition of such information. SIR also stores its information on the property lists of LISP atoms. The information from the first sentence would be stored by placing NOSE as a value of the property SUBPART of the atom PERSON, and the opposite relation (i.e. SUPERPART) on the atom NOSE. As the above conversation indicates SIR can go through many levels of property lists to discover the answer to a question.

Finally CARPS can be viewed as an extension of a symbolic manipulation program. A program such as CARPS would be much more difficult to write if it did not have this work to draw upon. In particular SCHVUOS, a simplification routine, and DIFF a differentiation routine, both written by J. Moses were used in this project. SCHVUOS and DIFF come from the routines SIN (Symbolic INtegrator), and SOLDIER (SOLution of Ordinary DIfferential Equations Routine) described in (6).

## II. STUDENT

As we mentioned earlier Bobrow's program, STUDENT, solves algebra word problems. It does this in the following manner:

1) Performance of mandatory transformations on the English sentences of the input. Labeling of word and symbols in text (such as verbs, delimiters).
2) Kernelization of sentences.
3) Transformation of kernelized sentences into equations.
4) Attempt at solution of equations. If successful then stop, else continue on.
5) Addition to equation set of possible pertinent equations from memory. Realization that two variables which were considered independent are actually the same variable.
6) Replacement of expressions by a possible alternate meaning.
7) Requesting more information from the user.

In the case of the last three stages an attempt is made to solve the equations after each stage, except for the last stage if the user has no more information to give.

8) For the sake of completeness, we must mention the fact that new information can be inserted into the store of global information available to the program. Such insertion is independent of the problem solving procedure outlined above.

However, in order to really understand how STUDENT works we should go through a problem and see how the program solves it. A typical problem which STUDENT could solve is:

```
(THE GAS CONSUMPTION OF MY CAR IS 15 MILES PER GALLON.
IF THE DISTANCE BETWEEN BOSTON AND NEW YORK IS 250
MILES, WHAT IS THE NUMBER OF GALLONS OF GAS USED ON A
TRIP BETWEEN NEW YORK AND BOSTON Q.)
```

In this problem we do not have any mandatory transformations, however many of the words will be tagged. (A typical mandatory transformation would be "twice" changed to "2 times".) After the words are tagged the problem would then look like:

```
(THE GAS CONSUMPTION (OF/OP) (MY/PRON)CAR (IS/VERB) 15
MILES PER GALLON(PERIOD/DLM) IF THE DISTANCE BETWEEN BOSTON
AND NEW YORK (IS/VERB) 250 MILES , (WHAT/QWORD) (IS/VERB)
THE NUMBER (OF/OP) GALLONS (OF/OP) GAS USED ON A TRIP BETWEEN
NEW YORK AND BOSTON (Q/DLM))
```

The next section of the program breaks the sentences into what Bobrow calls kernel sentences. In this problem the first sentence will not be changed, however, the second will. It will become (for the sake of convenience we will drop the tags)

```
(THE DISTANCE BETWEEN BOSTON AND NEW YORK IS 250 MILES.
WHAT IS THE NUMBER OF GALLONS OF GAS USED ON A TRIP BETWEEN
NEW YORK AND BOSTON Q.)
```

To understand exactly how this is accomplished we must know a little about METEOR, the language in which STUDENT was written. A METEOR program consists of a series of rules which are executed in sequence, subject to control statements in the rules. The left half of any rule is a pattern to be compared against the contents of the "workspace". (In our case the workspace would contain the sentence "If the distance...") If the pattern matches, the workspace is then changed to correspond to that once we have translated our sentences into equations, all algebra word problems are alike, that is, they are all sets of linear equations with as many equations as unknowns. Hence, once the problems has been reduced to equation form, STUDENT does not need any heuristics to solve the equations.

The second property is in my estimation even more serious. With one exception, each kernel sentence is translated into exactly one complete equation. As we shall show later (and it should not be too hard to convince oneself) this occurrence is not typical. The one exception in STUDENT to this rule occurs when we have a construction like: "A number is added to 18. This sum is 67." The first of the two sentences does not give a complete equation, but only, expressed in LISP notation, (PLUS (NUMBER) 18). However the "This" which starts the second sentence is the key to replace the part of the second sentence coming before the "is" by the equation fragment generated by the first sentence. So we get (EQUAL (PLUS (NUMBER) 18) 67), a complete equation.

Since STUDENT can count on its input always forming equations, there is no provision for any other form of information storage. However, in a typical Calculus word problem we might have a sentence like "A ship is traveling east.", or "Water is flowing into a conical funnel." In each case there is information which should be stored, but clearly the equation form would not be the appropriate storage medium.

Both of these criticisms stem directly from the type of problem Bobrow sets out to solve. There are other difficulties with STUDENT which were probably left unsolved because they were not critical to the operation of the program and the size of the program had already "a" and "the".

The program then tries to solve this set of equations and finds that it can not. It then prints out

```
(USING THE FOLLOWING KNOWN RELATIONSHIPS)
((EQUAL (DISTANCE)(TIMES (SPEED)(TIME)))
(EQUAL (DISTANCE((TIMES (GAS CONSUMPTION)(NUMBER OF GALLONS
OF GAS USED))))

(ASSUMING THAT)
((DISTANCE)IS EQUAL TO (DISTANCE BETWEEN BOSTON AND NEW YORK))

(ASSUMING THAT)
((NUMBER OF GALLONS OF GAS USED) IS EQUAL TO (NUMBER OF GALLONS
OF GAS USED ON TRIP BETWEEN NEW YORK AND BOSTON))

(ASSUMING THAT)
((GAS CONSUMPTION IS EQUAL TO (GAS CONSUMPTION OF MY CAR))
```

The equations are stored in a glossary under a key word. The first word of each variable (unless the variable starts with "number of", in which case "number of" is ignored) is looked up in the glossary and the corresponding equations pulled out. STUDENT then matches up the variables in the new equations with the variables already in the problem. To match up any two variables P1 and P2 we have the rule that if P1 appears later in the problem than P2 then P1 must be completely contained in P2 in the sense that P1 is a contiguous substring within P2. After having made these observations STUDENT again tries to solve the equations, succeeds, and prints out the

```
(THE NUMBER OF GALLONS OF GAS USED ON A TRIP BETWEEN NEW YORK
AND BOSTON IS 16.66 GALLONS)
```

A close examination of STUDENT shows that it has some properties which make it less general than one would like. The first of these is what is in the right half of the rule. The pattern which would match our sentence is (IF $, (1/QWORD) $ ). What this means is that the workspace will match the pattern if the first word is "if" followed by any number of arbitrary words (that's what the $ means) followed by a comma, followed by any word which is labeled "QWORD" followed by any number of words. Clearly our sentence will match this pattern. The righthand side of this rule would look something like ((2 (PERIOD/DLM 4 5) where the 2 refers to whatever was matched with the second object on the lefthand side. In this case it was the first $ (which matched with "the distance between Boston and New York"). The same goes for the 4 and 5.

STUDENT next transforms the simple sentences into equations. The general rule used here is that the word "is" is changed to an equal sign, and words like "times", "divide" are changed to their algebraic equivalents. STUDENT accepts seven different formats for questions. The specific manner in which a question sentence is changed into an equation will depend on the format. Roughly speaking the equation formed is an equality between a newly created atom and the quantity to which the question refers. Our problem will create the following equations

```
(EQUAL X00001 (NUMBER OF GALLONS OF GAS USED ON TRIP BETWEEN
NEW YORK AND BOSTON))

(EQUAL (DISTANCE BETWEEN BOSTON AND NEW YORK) (TIMES 250(MILES)))

(EQUAL (GAS CONSUMPTION OF MY CAR) (QUOTIENT(TIMES 15
(MILES))(TIMES 1(GALLON))))
```

With the exception of X00001 (the newly created variable) our variables come directly from the words of the problem, ignoring any occurances of reached the size of the available memory. These are: 1) A very limited subset of English grammar. STUDENT cannot handle something as basic to English as a dependent clause. In general one must be very careful in one's choice of sentences when addressing STUDENT. 2) A lack of sophisticated heuristics to determine the equality of variables previously considered independent. 3) A small knowledge of the "real world" and a rather inflexible manner of using that knowledge.

Looking at this second criticism we see that Bobrow takes long English phrases from the problem as his variables and makes no attempt to analyze their structure. For example in the problem we just looked at "THE NUMBER OF GALLONS OF GAS USED ON A TRIP BETWEEN NEW YORK AND BOSTON" was a single variable. To determine that two different phrases are equivalent Bobrow has two rules. 1) If two phrases are identical except that one has a group of words where the other has a pronoun they are considered equivalent. For example "the number of guns the Russians have" and "the number of guns they have" will be considered equal. 2) If the second phrase forms a contiguous block of the first, they are also considered equivalent. (We have already seen this rule in action.) There are many forms of paraphrasing that these rules will not cover ("the volume of the pile" and "the pile's volume"). Moreover, in many cases these rules will equate objects which we would not want equated ("street light" and "street"). A dramatic example of this situation is obtained by replacing the question in the problem above by "What is the number of gallons of gas used on a trip between Paris and Peking". STUDENT would have still answered - 16.66 gallons.

## III. AN OVERVIEW OF CARPS

In the first chapter we presented a problem which CARPS was able to solve. In this section we wish to give an overview of the techniques CARPS uses in solving a word problem. This will enable those who do not wish to wade through the fine details presented later to get a basic idea of the program's operation.

Let us use the following problem as an example.

```
(A LADDER 20.0 FEET LONG LEANS AGAINST A HOUSE/. FIND THE
RATE AT WHICH THE TOP OF THE LADDER IS MOVING IF ITS FOOT
IS 12.0 FEET FROM THE HOUSE AND MOVING AWAY FROM THE HOUSE
AT THE RATE 2.0 FEET PER SECOND/.)
```

The first section, as we mentioned previously, will perform several kinds of operations upon the input string. The words in the problem are examined one by one. If the word FOO is to be tagged it is just replaced by (FOO PART-OF-SPEECH). Some common phrases are changed to an arbitrary standard form. In our problem the phrase AT THE RATE was changed to AT RATE. Also the program will change any occurance of IS followed by a verb to the verb by itself. CARPS also noticed that the word LADDER was a key word indicating equations which might be needed later. The output of the first section is

```
(THE PROBLEM WITH TAGS ON IS)
 (((A (LADDER NOUN) 20.0 (FT UNIT) LONG (LEANS
VERB) AGAINST A HOUSE) (1.)) (((FIND QWORD)  (RATE RWORD)
ATWHICH THE TOP OF THE (LADDER NOUN) (MOVING VERB) IF (ITS PRON)
FOOT (IS VERB) 12.0 (FT UNIT) (FROM PREP) THE HOUSE AND (MOVING
VERB) (FROM PREP) THE HOUSE (AT PREP) (RATE RWORD) 2.0 (FT UNIT)
PER (SEC UNIT)) (2.)))
(THE PROBLEM TYPE IS)
DISTANCE
```

The next section will break up the sentences into simple sentences. This section is written in CONVERT. A CONVERT program, like a METEOR program, basically consists of a series of rules, each specifying a pattern which the input must match if the rule is to be used, and instructions as to what the program should do if the pattern is matched.

For example our second sentence matches the CONVERT pattern which essentially looks for a sentence of the form:

QUESTION WORD - ANYTHING - IF - ANYTHING

| | | |
|---|---|---|
| SENTENCE: | FIND | RATE ATWHICH THE TOP OF THE LADDER MOVING |
| PATTERN: | QUESTION WORD | ANYTHING |
| SENTENCE: | IF | ITS FOOT IS 12.0 FT FROM THE HOUSE AND MOVING FROM THE |
| PATTERN: | IF | ANYTHING |
| SENTENCE: | HOUSE AT RATE 2.0 FT PER SECOND. | |

The rule then states that the sentence should be broken up into two sentences and the program restarted on each sentence. So we get the two sentences FIND RATE ATWHICH THE TOP OF THE LADDER MOVING. and ITS FOOT IS 12.0 FT FROM THE HOUSE AND MOVING FROM THE HOUSE AT RATE 2.0 FT PER SEC.

The first of these sentences cannot be broken up any further. The second however will match the pattern

ANYTHING - VERB - ANYTHING - AND / WHILE - VERB - ANYTHING

| | | | | | |
|---|---|---|---|---|---|
| SENTENCE: | ITS FOOT | IS | 12.0 FT FROM THE HOUSE | AND | MOVING |
| PATTERN: | ANYTHING | VERB | ANYTHING | AND /<br>WHILE | VERB |
| SENTENCE: | AWAY FROM THE HOUSE AT RATE 2.0 FT PER SEC. | | | | |
| PATTERN: | ANYTHING | | | | |

The rule here is to break the sentence in two, each having a noun phrase (matched by the first ANYTHING) as its subject, but a different predicate. Our sentence then becomes ITS FOOT IS 12.0 FT FROM THE HOUSE. and ITS FOOT MOVING AWAY FROM THE HOUSE AT RATE 2.0 FT PER SEC. The first of these is completely simplified. The second can once more be broken up, but we have already seen enough to get the basic idea of the process. After simplification the program prints.

```
(THE SIMPLIFIED SENTENCES ARE)
(((A (LADDER NOUN) (LEANS VERB) AGAINST A HOUSE) (1.))
((A (LADDER NOUN) (IS VERB) 20.0 (FT UNIT) LONG) (1.))
(((FIND QWORD) (RATE RWORD) ATWHICH THE TOP OF THE
(LADDER NOUN) (MOVING VERB)) (2.))
(((ITS PRON) FOOT (IS VERB) 12.0 (FT UNIT) (FROM PREP)
THE HOUSE) (2.))
(((ITS PRON) FOOT (MOVING VERB) (FROM PREP) THE HOUSE) (2.))
(((ITS PRON) FOOT (MOVING VERB) (AT PREP) (RATE RWORD) 2.0
(FT UNIT) PER (SEC UNIT)) (2.)))
```

Once the sentences are simplified the program goes to the third and probably most important stage. It is here that we translate our simple sentences into the internal representation of the problem (i.e., into structures and equations). This takes place is two phases. Let us call them A and B. The first phase identifies the basic form of a sentence. The second phase further analyzes the sentence components recognized by phase A, and relates the sentences to the entire problem of which it is a part. Phase B concludes by filling out the structures. The entire procedure is applied to each sentence.

The first sentence does not match any of the basic patterns used in building structures. The second sentence matches the pattern

| | | | | |
|---|---|---|---|---|
| A LADDER | IS | 20 | FT | LONG |
| NVP | VERB | NUMBER | UNIT | SINGLE ATOM |

NVP stands for No Verb Phrase. This means that it will match any group of

[redacted]

The LENGTH node is included just to remind us that it is still there. The equation generated is:

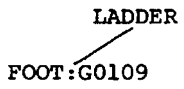
(EQUAL (G0108) (DERIV(G0107 G0106 LADDER)))

G0108 is just a newly created variable. The expression (G0107 G0106 LADDER) is the variable which represents the position of the top of the ladder. Note that G0106 is just the value of TOP, as G0107 is the value of POSITION.

Since this section is so important, let us go through one more example before going on the section 4. The next sentence matches the pattern

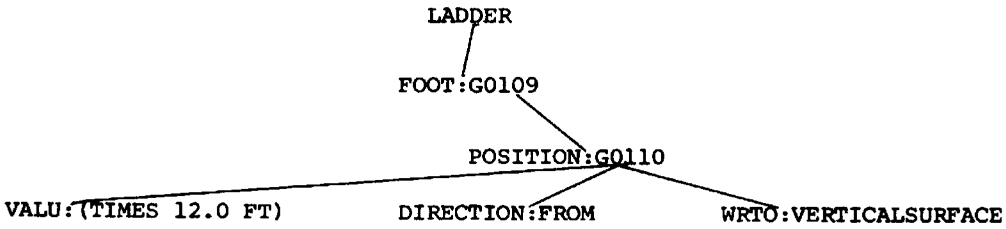
| | | | | | |
|---|---|---|---|---|---|
| ITS FOOT | IS | 2.0 | FT | FROM | THE HOUSE |
| NVP | VERB | NUMBER | UNIT | POSITIONAL | NVP |

A "positional" is a specially created class of adverbs and prepositions which indicate a positional relationship between two nouns. So this type of basic sentence is concerned with the position of one object with respect to another. The first noun phrase becomes



The possessive pronoun is assumed to refer to the top level of the previous structure of the correct type. That is, ITS will refer to the previous nonhuman structure mentioned while HIS would refer to the previous human structure mentioned.

To this is added the fact that the foot of the ladder is 2.0 ft from the house.



The POSITION node is put on because the sentence is one which basically deals with position. VERTICALSURFACE is just a general name for house in this context (also for fence, wall, etc.). In the same manner the rest of the sentences are translated into structure form. The final structure is shown on the next page (figure 3).

The fourth section retrieves pertinent equations from memory and replaces the variables in the equations with specific references to the structures. This section notes that the problem type is DISTANCE. Looking on its keyword list it notes that the only keyword is LADDER. This causes it to pull the following equation out of memory

```
(EQUAL (EXPT (LENGTH OBJ)2.0)
       (PLUS (EXPT(POSITION TOP OBJ)2)
             (EXPT (DISTANCE (FOOT OBJ)(VERTICALSURFACE))2))
```

This is just

$$\text{LENGTH}^2 = \text{HEIGHT}^2 + \text{DISTANCE TO VERTICAL SURFACE}^2$$

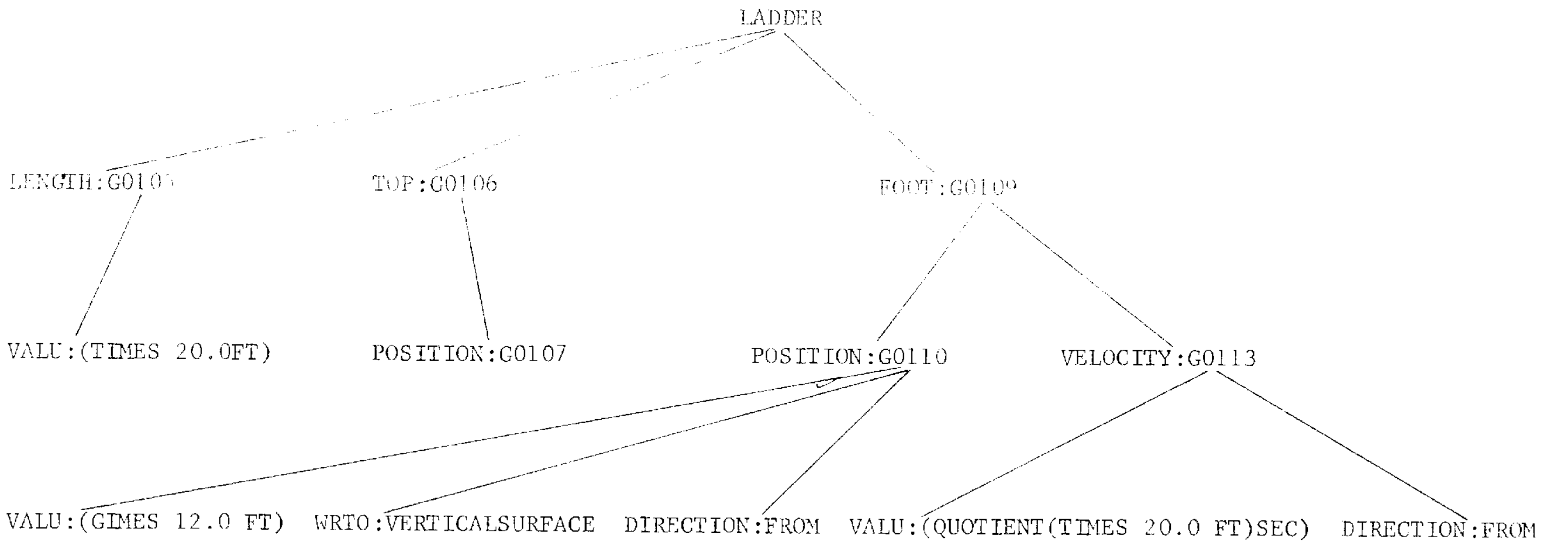

LADDER
LENGTH:G0105
TOP:G0106
FOOT:G0109
VALU:(TIMES 20.0FT)
POSITION:G0107
POSITION:G0110
VELOCITY:G0113
VALU:(GIMES 12.0 FT)
WRTO:VERTICALSURFACE
DIRECTION:FROM
VALU:(QUOTIENT(TIMES 20.0 FT)SEC)
DIRECTION:FROM

Fig. 3

Also the variable OBJ is set to LADDER. (Note that the problem could have had the word "pole" or "rail" for that matter. The same equation would have been used, but OBJ would have been set to the respective word.)

Now the program goes through the equation and replaces the variables with the information from the structures. the expression (POSITION TOP LADDER) is replaced by (G0107 G0106 LADDER). We have replaced the names of the markers by their VALU in the structure. The next variable is replaced by (TIMES 20 FT). The variable (DISTANCE (FOOT LADDER) (VERTICALSURFACE)) is replaced by an expression for the distance. The final replaced equations are:

```
(THE EQUATION SET IS)
 1((EQUAL (G0108) (DERIV (G0107 G0106 LADDER)))
 2(EQUAL (EXPT (G0107 G0106 LADDER) 2.) (PLUS
 (TIMES 400.0 (EXPT FT 2.)) (TIMES -1. (EXPT (
 PLUS (TIMES 12.0 FT) (TIMES 2.0 T1M FT (EXPT
 SEC -1.))) 2.)))))
```

We now move to section five, which manipulates the final equations. Before the program can solve for (G0108) in equation one, it must first find an expression which relates (G0107 G0106 LADDER) to time, and contains no other variables. Equation 2 satisfies this criterion, so it is solved for (G0107...) and differentiated with respect to time. In this respect this problem is an easy one for CARPS. Other problems would require more equation manipulation before a suitable expression relating one of the variables to time could be found. It then notes that no boundary conditions were mentioned, so it assumed time = 0, substitutes this into the equations, simplifies, and prints

```
(THE ANSWER IS)
(TIMES -1.5 (EXPT SEC -1.) FT)
```

Which in more standard notation is -1.5 FT / SEC.

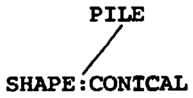
Before we move on to a detailed exposition of CARPS, we would like to make explicit a few points which did not come out in the previous discussion.

We have chosen to break sentences up into smaller sentences. There is some theoretical justification for doing so, that is the work of the transformationalists in linguistics. However the primary reason was that a sentence is the most self-contained small unit of information which is currently available. While it is true that the individual sentence is not always self-contained, (e.g., a sentence containing a pronoun or understood object) it is frequently so; the exceptions can be treated individually.

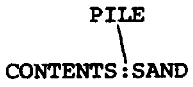
To a limited extent CARPS can use knowledge of the real world to parse a sentence. Its knowledge of cones (see the example of chapters four, five and six) enables it to recognize in a problem which deals with cones that THE RADIUS refers to THE RADIUS OF THE BASE OF THE CONE.

We decided on the use of structures for several reasons. Given the property lists and the functions for manipulating them in LISP, structures are a natural form of information storage.

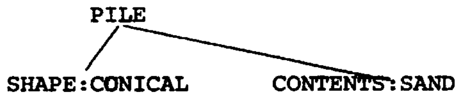
Together with deeper grammatical analysis of noun phrases, structures also help us go a long way toward solving the paraphrase problem which Bobrow encountered. Suppose the phrases CONICAL PILE and PILE OF SAND were encountered in a single problem. STUDENT would not realize that they might be the same object. CARPS however would analyze the first as



and the second as



which would give us



Finally structures allow us to store interrelated information according to their relationships. Though this has not been shown clearly so far, the problem we will analyze in the next section will show exactly how useful structures are in this respect.

## IV. WORD TAGGING AND SENTENCE DECOMPOSITION

If we are to really understand the workings of CARPS we should go through a problem in great detail. Consider the following problem:

> (WATER IS FLOWING INTO A CONICAL FILTER AT THE RATE OF 15.0 CUBIC INCHES PER SECOND/. IF THE RADIUS OF THE BASE OF THE FILTER IS 5.0 INCHES AND THE ALTITUDE IS 10.0 INCHES/, FIND THE RATE AT WHICH THE WATER LEVEL IS RISING WHEN THE VOLUME IS 100.0 CUBIC INCHES/.)

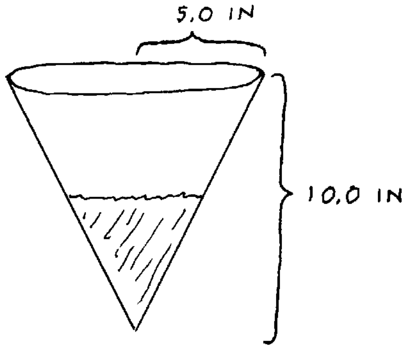


Fig. 4

In the first section of the program each word is checked to see if it has a property on its property list under the indicator GRAMMAR. If it does the value of this property will be a function, which will then be evaluated. The net result of the evaluation will be one of the following.

1) The current word (i.e., the word whose property list we just checked) will be tagged with its part of speech.

2) If the word under consideration is a question word (such as FIND or HOW) the ensuing words will also be checked to see if they give any clue as to the type of problem we are dealing with. That is, the property lists of the words will be checked for the identifiers

PTYPE and GRAMMAR and any value noted. The logic behind this action is that the most reliable place to look for clues to the problem type is where the information is actually requested, i.e., in the question. This action will be terminated when the sentence ends or when a word is encountered which indicates that the rest of the sentence is not part of the question proper. In our problem FIND is the question word, and the word LEVEL, which in this context usually means altitude, is the clue that the problem is one which deals with volumes. The search for such clues was ended in this problem upon the encounter of the word WHEN.

3) The current word is noted as a key word (meaning it has equations associated with it in memory), the sentence however is left unchanged unless, of course, the word is also to be tagged with a part of speech. (The only key word occuring in our problem is CONICAL.)

4) The word may be the signal word for a mandatory transformation in which case the local context of the word is checked and if correct the transformation is performed. In this problem we have several of these transformations. The phrases AT THE RATE OF and AT THE RATE are changed to AT RATE. (The signal word here is RATE.) Cubic inches is changed to IN3, AT WHICH is changed to ATWHICH. There are many others as a glance at the output will show.

5) At the end of a sentence (as indicated by a "?", ".", or ";") the end punctuation is deleted, and the sentence given a tag with a number indicating its order in the problem.

If a word has no GRAMMAR on its property list it will be checked for two conditions. If it ends in ING it will be labeled VERB. If it ends in LY it will be labeled ADV. Though there are many exceptions to these rules the limited vocabulary encountered in these types of problems does not include any that we have been able to find.

After this section is through, our problem will look like this:

```
(((WATER (FLOWING VERB) (INTO PREP) A (CONICAL ADJ) FILTER (AT
PREP) (RATE RWORD) 15.0 (IN3 UNIT) PER (SEC UNIT)) (1)) ((IF THE
RADIUS OF THE BASE OF THE FILTER (IS VERB) 5.0 (IN UNIT) AND THE
ALTITUDE (IS VERB) 10.0 (IN UNIT) /, (FIND QWORD) (RATE RWORD)
ATWHICH THE WATER LEVEL (RISING VERB) WHEN THE VOLUME (IS VERB)
100.0 (IN3 UNIT)) (2)))
```

At this point the program looks over the list of words which it has accumulated to indicate the problem type. If the majority indicate one type of problem, the indicator PROBTYPE is set to that name. Otherwise the program will print an error message and halt.

The next section of the program breaks the sentences into simple sentences. Since we treat this as a problem of pattern matching, it is written in CONVERT. At the present moment we have sixteen rules in this part of the program. They cover a large portion of compound and complex sentences. These rules are listed in Table 1.

Though we will not use actual CONVERT notation it would be useful if one had a slight idea of how the language operated.

CONVERT is used like any other LISP function, that is

```
(LAMBDA (ARG1 ARG2....) (CONVERT  M E I R))
```

It has four arguments, as indicated by the M E I R. The first two are

## TABLE 1: SECTION 2

| PATTERN | EXAMPLE | USE |
|---|---|---|
| IF - ANYTHING - , - QUESTION WORD - ANYTHING | If - the radius is 3 ft - , - what - is the altitude | To break up sentences which contain a subordinate clause introduced by IF |
| QUESTION WORD - ANYTHING - IF - ANYTHING | What - is the altitude - if - the radius is 3 ft | Same, except that clause occures at the end of the sentence |
| QUESTION WORD - ANYTHING - WHEN - ANYTHING | What - is the altitude - when - the radius is 3 ft | Separate off from the sentence a subordinate clause, introduced by WHEN |
| WHEN - ANYTHING - QUESTION WORD - ANYTHING | When-the radius is 3 ft-what-is the altitude | Same, except that clause occurs at the beginning of the sentence |
| ANYTHING - VERB - ANYTHING - AND/WHILE - VERB - ANYTHING | The radius - is - 3 ft - and - increasing - at 3 ft per sec | To break up a sentence which contains two predicates, connected by AND or WHILE |
| ANYTHING - VERB - ANYTHING - AND/WHILE - ANYTHING - VERB - ANYTHING | The radius - is - 3 ft - and - the altitude - is 5 in | To separate two complete sentences joined by AND or WHILE |
| ANYTHING - WHOSE - ANYTHING - VERB - ANYTHING - VERB - ANYTHING | The cone - whose - radius - is - 3 ft - has - an altitude of 5 ft | To separate a subordinate clause introduced by WHOSE |
| ANYTHING - VERB/PREP - RP - WHOSE - ANYTHING - VERB - ANYTHING | A man walks - toward - a light - whose - height - is 20 ft | Same, except that clause occurs at the end of the sentence |
| ANYTHING - VERB/PREP - RP - WHICH/THAT - VERB - NVP | A man walks toward - a light - which - is - 20 ft high | To separate a subordinate clause introduced by WHICH or THAT |
| RP - , - ANYTHING - , - VERB - ANYTHING | A man - , - starting at 5 PM - , - walks toward a light | To separate an appositive set off by commas |
| NVP - VERB - NVP - , - NVP - , - NVP | A box - is - 10 ft long - , - 3 ft wide - , - and 4 ft high | To separate a sentence with 3 objects or predicate adjectives |
| NVP - VERB - ADVERB - ANYTHING | A man - walks - directly - toward a light | To create a new sentence with the adverb and verb as its predicate |
| NVP - VERB - PREP - RP - PREP/ADVERB - ANYTHING | A man - walks - toward - a light - at - 2 miles per hour | To break up 2 predicate phrases, the first starting with a prep, the second with a prep or adverb |

used to define the variables of the program. The third is the list which is the input to the program. If the LAMBDA has more than one argument they must be made into a list by the argument I (i.e., I would be (LIST A B) ) as the final argument R, the list of rules, will only accept one list as its input.

R is composed of one or more sections, each of which has a name and consists of one or more rules. In a given section the input is matched against the left half of the rule. If it matches, the value of the function is just the value of the righthand side of the rule. Should none of the patterns in the section match, the value of the function is just the unmatched expression. If a rule does apply, however, the righthand side may give control information, like "the value of this expression is just the value of the entire program started over, on the expression '(A X Q D)'". This is just a recursive call. If none of the patterns in a given section match the expression the value of the program in just the input expression.

Let us start with the second sentence, since it is somewhat easier to explain than the first. The pattern which will match this is

IF - ANYTHING - , - QUESTION WORD - ANYTHING

```
IF   THE RADIUS OF THE BASE OF THE FILTER IS 5.0 IN AND THE ALTITUDE
IF                            ANYTHING
IS 10.0 IN     ,      FIND   RATE ATWHICH THE WATER LEVEL RISING WHEN
             ,  QUESTION WORD                ANYTHING

THE VOLUME IS 100.0 IN3
```

The right side of the rule specifies that we begin the program over, but break the sentences in two. In our case the two sentences are THE RADIUS OF THE BASE OF THE FILTER IS 5.0 IN AND THE ALTITUDE IS 10.0 IN and FIND RATE ATWHICH THE WATER LEVEL RISING WHEN THE VOLUME IS 100 IN.

The first of these sentences will then match the rule;

ANYTHING - VERB - ANYTHING - AND/WHILE - ANYTHING - VERB - ANYTHING

| | | | |
|---|---|---|---|
| THE RADIUS OF THE BASE OF THE FILTER | IS | 5.0 IN | AND |
| ANYTHING | VERB | ANYTHING | AND/WHILE |

| | | |
|---|---|---|
| THE ALTITUDE | IS | 10.0 IN |
| ANYTHING | VERB | ANYTHING |

This rule matches the case where an AND or a WHILE connects two complete sentences. It then recurses on each of the two sentences separately. The two sentences formed in this way (i.e., THE RADIUS OF THE BASE OF THE FILTER IS 5.0 IN and THE ALTITUDE IS 10.0 IN) will not match any more patterns so they will remain in their present form.

The second sentence we mentioned above (FIND THE RATE ATWHICH...) will be broken up by the rule

QUESTION WORD - ANYTHING - WHEN - ANYTHING

| | | |
|---|---|---|
| FIND | RATE ATWHICH THE WATER LEVEL RISING | WHEN |
| QUESTION WORD | ANYTHING | WHEN |

| |
|---|
| THE VOLUME IS 100.0 IN3 |
| ANYTHING |

This rule is fairly straight forward in the sense that it splits off the dependent clause introduced by WHEN and makes it a separate sentence. However, in the process the WHEN is removed from the sentence. This action is more important than it might first appear. The word WHEN in calculus problems is almost always used to indicate the boundary condition. That is, it precedes a value which should only be substituted into the equations after the differentiation is accomplished. A good illustration of this would be the sentence "At what rate is the radius increasing when the radius is 3 feet.". Clearly if we substituted in the value first we would get zero no matter what the rest of the problem said. In order to save this information the word WHEN is added to the tag of the second sentence created by this rule. The two sentences formed by this rule (THE RATE ATWHICH WATER LEVEL RISING and THE VOLUME IS 100.0 IN3) cannot be broken up any further.

Let us now return to the first sentence. The rule that will match here is the following:

NOUN PHRASE - VERB - PREP - PHRASE - PREP / ADV - PHRASE

| | | | | | |
|---|---|---|---|---|---|
| WATER | FLOWING | INTO | A CONICAL FILTER | AT | RATE IS |
| NVP | VERB | PREP | RP | PREP | |

IN3 PER SEC
ANYTHING

RP stands for Restricted Phrase, which is a non-null string of words which contains no verbs, prepositions, or adverbs. This rule breaks our sentence into two again, each having one of the two original phrases which began with prepositions. So we get WATER FLOWING INTO A CONICAL FILTER and WATER FLOWING AT RATE 15.0 IN3 PER SEC.

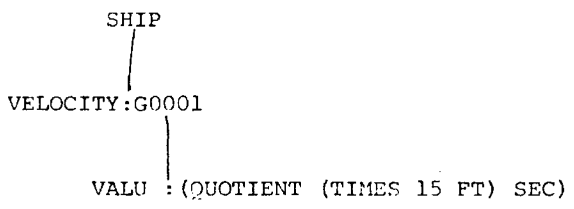
With this transformation our sentences are completely broken up as far as the program in concerned. Our problem now looks like (with the word tagged just to remind us that the tags are still there)

```
(((WATER (FLOWING VERB) (INTO PREP) A (CONICAL ADJ) FILTER) (1))
((WATER (FLOWING VERB) (AT PREP) (RATE RWORD) 15.0 (IN3 UNIT) PER
(SEC UNIT)) (1)) ((THE RADIUS OF THE BASE OF THE FILTER (IS VERB)
5.0 (IN UNIT)) (2)) ((THE ALTITUDE (IS VERB) 10.0 (IN UNIT)) (2))
(((FIND QWORD) (RATE RWORD) ATWHICH THE WATER LEVEL (RISING
VERB)) (2)) ((THE VOLUME (IS VERB) 100.0 (IN3 UNIT)) (2  WHEN)))
```

## V. TRANSFORMATION OF SENTENCES INTO THE INTERNAL MODEL

We now come to the main section of the program. It is in section 3 that we try to abstract the information in the sentences, and place this information conveniently in our model.

Before we go into the workings of this section, let us take another look at the representation of structures.



We see that we have put the tag VELOCITY on the atom SHIP with the value G0001. Then on the property list of this atom we put the property VALU with the value (QUOTIENT (TIMES 15 FT) SEC).

The value of the velocity is not placed on the property list of SHIP directly for two reasons. One is based on detailed programming considerations. The second is the fact that we may wish to put more information concerning the velocity into the structure, such as the direction of the velocity. It seemed to us most logical to have this information also hanging from the VELOCITY node. This requires that the value of velocity be an atom from which we can hang more information. This atom must be a gensym and not an arbitrary label. If it were the latter, and we put the value of another velocity into our model we would erase the first value unless we took some rather complex precautions.

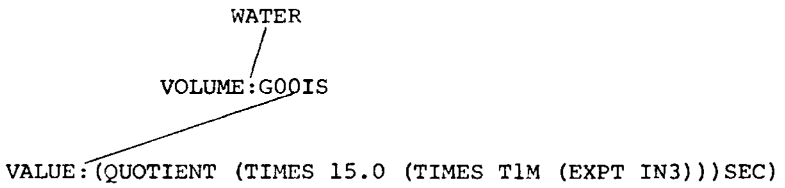
Upon entering phase A with the first sentence we check for two conditions. If the word WHEN is in the tag, then a flag is set. We then check to see if the sentence is a question. In this case the sentence is routed to a special section which deals only with questions. If not, it will go to one of two possible sections depending on the problem type (since certain types of problems tend to have preferred types of sentences). However if no match is found in the first section, the other will be tried. A list of the various classifications of sentences in phase A is found in Table 2.

The first sentence will not match any pattern and will be placed on the special list for unmatched sentences. The second sentence will match the pattern

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| WATER | FLOWING | AT | RATE | 15.0 | IN3 | PER | SEC |
| NVP | VERB | PREP | ANYTHING | NUMBER | UNIT | PER | TIMEUNIT |

This format indicates that the object mentioned by the noun phrase is changing at a given rate. The rate is restricted to time simply because all our problems have only time rates.

We now enter phase B. It notices that IN3 is a unit of volume, so we get the structure:

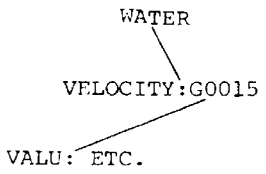


## TABLE 2: SECTION 3, PHASE A

| PATTERN | EXAMPLE | COMMENTS |
|---|---|---|
| <u>VOLUME SECTION</u> | | |
| NVP - VERB - PREP - ANYTHING - NUMBER - UNIT - PER - TIME UNIT | A ship - traveling - at 20 - mile - per - hour | Indicates something changing at a constant rate. |
| NVP - IS - NUMBER - UNIT | A radius - is - [illegible] - in | The basic value assignment statement. |
| NVP - IS - NUMBER/NUMBER "TIMES"/NUMBER "OF" - NO UNIT - NVP | The altitude - is - .5 times - the - radius of the base | This will create an equation. |
| NVP - IS - NUMBER - UNIT - SINGLE WORD | The cone - is - 8 - in - tall | |
| <u>DISTANCE SECTION</u> | | |
| NVP - VERB - POSIT | A ship - traveling - north | Indicates direction of velocity. |
| NVP - VERB - AT - TIME INDICATOR | A ship - starting - at - 12 AM | Indicates zero time. The time indicator may also be "noon" or "midnight". |
| NVP - VERB - NUMBER - UNIT - POSIT - NVT | A ship - is - 30 - mile - north - point 0 | Indicates value and direction of position with respect to a second object. |
| NVP - VERB - POSIT - NVP | A ship - is - north - point 0 | Indicates direction of position with respect to a second object. Only used in distance problems. |
| <u>QUESTION SECTION</u> | | |
| QUESTION WORD - RWORD - ANYTHING - VERB - RP/NULL - TIME INDICATOR - RP/NULL | Find - rate - atwhich they - separating - at - 3 PM - - | Boundary conditions explicitly involving time have different forms, this rule covers most. |
| QUESTION WORD - RWORD - ANYTHING - VERB - OR - VERB | What - rate - are they - approaching - or - separating | |
| QUESTION WORD - RWORD - "ATWHICH"/VERB - NVP - VERB | How - fast - is - the altitude - rising | This is the basic rate question format. |

TIM is the internal symbol for time. Notice that the rate has been automatically integrated to a volume.

If we had "Water flowing at 15.0 MILES per hour." we would have created instead:



The program now realizes that this sentence indicates volume change. It will then go and look at the previous unused sentences to see if it can find one of the form, (WATER FLOWING-preposition-nounphrase) This would indicate that water is the contents of whatever corresponds to the noun phrase. This indeed matches our first sentence with noun phrase = (A CONICAL FILTER). Phase B then calls the program, PVA, which analyzes noun phrases. In Table 3 we have a list of the operations PVA can perform. PVA, after applying a few tests which fail, strips the phrase of all occurences of A and THE. It then notes that conical is an adjective and that SHAPE is the value of the property ADJTYPE on its property list. Then this information, as well as the contents of the filter, is put into structure form and we get:

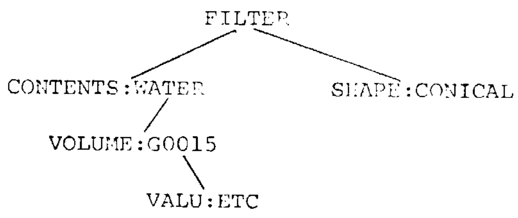


TABLE 3: PVA

| PATTERN | EXAMPLE | | COMMENT |
|---|---|---|---|
| ANYTHING - OF - A/THE/PRONOUN - ANYTHING | the radius - of - the - base | BASE<br>RADIUS:G0001 | |
| ANYTHING - OF - NO NUMBER | a pile - of - sand | PILE<br>CONTENTS:SAND | Only used in volume problems |
| ANYTHING - DISTANCE - FROM - | the train's - distance - | TRAIN<br>POSITION:G0001<br>DIRECTION:FROM WRTO:SHIP | |
| ANYTHING - DISTANCE - FROM/BETWEEN - ANYTHING - TO/AND - ANYTHING | The - distance - between - the train - and - the ship | as above | |
| POSSESIVE PRONOUN - ANYTHING | its - altitude | FILTER<br>ALTITUDE:G0001 | Filter was the name of the previous non human structure mentioned in the problem |
| SINGULAR PRONOUN | he , she , it | | Return the name of the last human or nonhuman structure mentioned |
| PLURAL PRONOUN | they | | Return names of last two different structures mentioned |
| ADJECTIVE - ANYTHING | red balloon | BALLOON<br>COLOR:RED | |

Note that PVA is applied to all noun phrases. In the second sentence it was applied to WATER, but naturally it was not able to break it up further. The fact that atoms are unique in LISP comes in very handy here. When we put WATER on the property list of FILTER we are guaranteed that all the properties associated with WATER (VOLUME in this case) will tag along.

The third sentence (THE RADIUS OF THE BASE OF THE FILTER IS 5.0 IN) will match

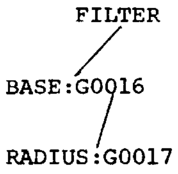
| | | | |
|---|---|---|---|
| THE RADIUS OF THE BASE OF THE FILTER | IS | 5.0 | IN |
| NOUN PHRASE | IS | NUMBER | UNIT |

PVA is applied to THE RADIUS OF THE BASE OF THE FILTER. It first notices that this is of the form

(THE RADIUS)OF(THE BASE OF THE FILTER)

However the right hand side of the above, (THE BASE OF THE FILTER) is also in this form, so we first analyze it, which gives us

We then analyze the entire phrase giving



Finally PVA returns this structure and on the property list of G0017 is put VALU:(TIMES 5.0 IN).

The fourth sentence (THE ALTITUDE IS 10.0 IN) matches the same form. Since the phrase THE ALTITUDE cannot be broken up any further, PVA analyzes it as a single level structure made up of the single atom ALTITUDE.

To understand how the fourth sentence is handled we must backtrack somewhat. When we encountered the phrase CONICAL FILTER we put conical on the property list of filter. The program knows that conical objects have many properties in common. For example they all have altitudes and bases, and the latter has a radius. So these facts are put on the structure of filter. Hence much of the structure we have shown being created had already been placed on the structure as a common property of cones. Nevertheless at each step the system actually did create the substructure finding, however, that the information was already on the property lists.

There is one other operation we have so far neglected. Every time a property X is put on the atom Y, a property MARKERON is placed on the atom X, and this has the value Y. This gives us backwards pointers so that given a property we can find out which atoms have that property. Furthermore if the value of Y is Z, and Z is an atom, on the property list of Z will appear VALUON:(X Y). .If Z is the value of more than one x,y pair all such pairs will be listed. Since we had already created the substructure

```
       FILTER
          \
ALTITUDE:G9002
```

on the property list of altitude was a pointer to FILTER. Hence when PVA returned the single level structure ALTITUDE, the program checked to see if ALTITUDE was on the property list of any other atom. Finding that it was a marker on FILTER the program assumes that ALTITUDE refers to the altitude of the filter, and creates the substructure

```
       FILTER
          \
ALTITUDE:G9002
            /
      VALUE:(TIMES 10.0 IN)
```

The next sentence is the question (FIND RATE ATWHICH THE WATER LEVEL RISING). Since we have illustrated this question format already in chapter one we will not do so again. PVA again analyzes the noun phrase WATER LEVEL and gives

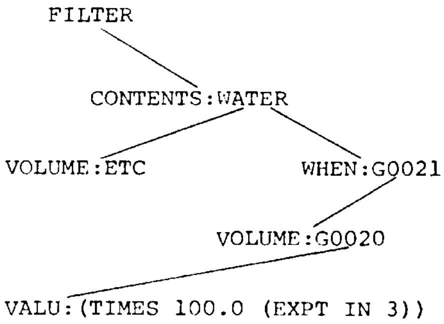
```
       WATER
          \
ALTITUDE:G0019
```

It does this because the program knows that LEVEL in this context "(noun phrase) LEVEL" means ALTITUDE. (This information is stored on the property list of level.)

Once again WATER is only on the property list of FILTER, so we can trace back and find that what we are interested in is (G0019 WATER FILTER). This is the equation variable format. This variable represents, in English, the altitude of the contents of the filter. So we create the equation

(EQUAL G0020 (DERIV(G0019 WATER FILTER)))

The interpretation of the last simple sentence (THE VOLUME IS 100.0 IN3) depends on the same sort of analysis. The word volume only appears as a marker on water, so the program assumes the sentence is refering to this volume. However since the tag for this sentence has the word WHEN, the structure created is



The WHEN on the property list of WATER indicates that what follows beneath it is a boundary condition.

At this point all the necessary information has been extracted from the sentences, and this stage of the solution is over. The final structure is illustrated on the next page.


VALU:(QUOTIENT(TIMES 1.0 (TIMES 1.M (EXPT IN 3))SEC)
VOLUME:000
VALU:(TIMES 0 IN)
VALU:(TIMES 100.0 (EXPT IN 3))

Fig. 5

## VI. THE FORMATION OF THE EQUATIONS AND THEIR SOLUTION

The fourth section is concerned with establishing the equations which the fifth section will manipulate to solve the problem. The program first checks the problem type. Finding that it is VOLUME it then checks the keyword list, which has the single element CONICAL. The property list of CONICAL is now checked. By this time CONICAL has the backwards pointer VALUON(FILTER SHAPE) which tells us that filter is our only conical object. We then pull out the equations connected with the word conical. We also set the variable OBJ to FILTER so we will know exactly what is conical. Note that rather than talk about a conical filter we could have mentioned a "cone". This would cause no problem since the word cone is defined in our dictionary by

```
      CONE
      /
SHAPE:CONICAL
```

(Actually it is defined by a LISP function which says that when cone is encountered in the right context, one should give it the properties of a conical object. Otherwise we would always have a backwards pointer from CONICAL to CONE even in problems which did not mention cones.)

In the case of distance problems, a routine is always called which can set up a distance equation between two objects, given their names. It does this with the information stored in the structures of the two objects. While finding the velocity and position of each object it notes their direction. In the case of velocity it also notes at what time the motion began (if no time is specified it assumes t = 0). The program checks to see if all positions are measured with respect to the same co-ordinate system, returning an error message if not. It then collects the position and velocity vectors in the same direction into a single term. It will return the square root of the sum of the squares of the terms. The program is able to handle some cases of the equivalence between the distance from an object to the ground (or street) and the altitude of that object.

There will be two equations (other than the one already created by the question sentence). One relates the volume of the contents of a cone to the altitude and radius, and the other relates the height and radius of our cone to the height and radius of the contents of the cone. Since the equations are stored in memory all parameters are represented by lists which specify what piece of information is acceptable as the value of this parameter. For example, if the parameter represented the altitude of some object, its representation would be (ALTITUDE OBJ). Before these equations can be used, these lists must be replaced by their values. Consider this equation used in this problem:

```
(EQUAL (TIMES (RADIUS BASE CONTENTS OBJ)
              (ALTITUDE OBJ))
       (TIMES (ALTITUDE CONTENTS OBJ)
              (RADIUS BASE OBJ)))
```

This is the relationship, R / CR = A / CA as seen in the diagram

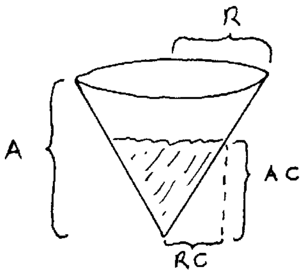


Fig. 6

(For a list of equations which CARPS has available as well as examples of other types of information CARPS knows about certain words see Table 4.)

Before the parameters are replaced an extra copy is set aside to be used later when we solve for the boundary condition. The program will then go through the list structure noting that EQUAL and TIMES are algebraic symbols. It will then come to (RADIUS... ) and note that this is not an algebraic symbol, implying that the list of which RADIUS is the first element is a variable name. Processing the variable in reverse order, it first finds that the value of OBJ is FILTER. It then looks for the contents of FILTER, which is WATER, and then for the BASE of the WATER, which it does not find. Hence it leaves this variable with only the last two elements replaced (e.g., (RADIUS BASE WATER FILTER)). It will find the altitude of the filter however, and note that it has the property VALU whose value will then replace this entire expression in the equation. In this same manner the rest of the equation set will be filled, and the computer prints out:

```
(THE EQUATION SET IS)
1 ((EQUAL (G0005) (DERIV (G0004 WATER FILTER)))
2 (EQUAL (QUOTIENT (TIMES 17.0 (TIMES (EXPT IN 3) T1M)) SEC)
  (TIMES (G0004 WATER FILTER) 0.33333300 PI
  (EXPT (RADIUS BASE WATER FILTER)2)))
3 (EQUAL (TIMES (RADIUS BASE WATER FILTER) (TIMES 12.0 IN))
  (TIMES (G0004 WATER FILTER) (TIMES 5.0 IN))))
```

$$\text{G0005} = \frac{d}{dt}\ \text{ALTITUDE}$$

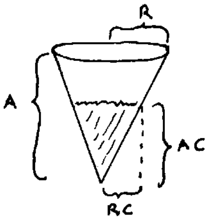
$$(17\ \text{IN}^3/\text{SEC})t = \frac{1}{3}\pi\ \text{RADIUS}^2\ \text{ALTITUDE}$$

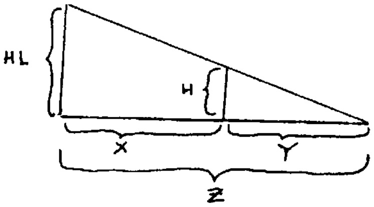
$$12\ \text{IN} * \text{Radius} = 5\ \text{IN} * \text{ALTITUDE}$$

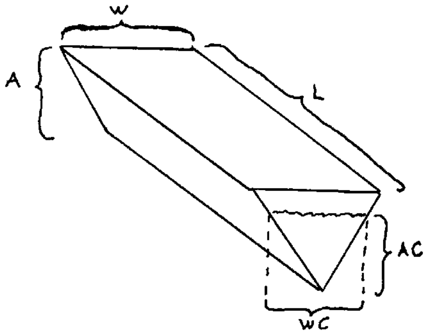
# TABLE 4: KNOWLEDGE ABOUT WORDS

## KNOWLEDGE IN THE FORM OF EQUATIONS

| | |
|---|---|
| conical | $VOLUME\ OF\ CONTENTS = 1/3 * \pi * RC^2 * AC$<br>$RC * A = AC * R$ |
| disk | $AREA = 2 * \pi * R^2$<br>$VOLUME = \pi * R^2 * THICKNESS$ |
| ladder | $L^2 = X^2 + Y^2$ |
| shadow | $Y * (HL - H) = H * X$<br>$Z = X + Y$ |
| spherical | $VOLUME = 4/3 * \pi * R^3$<br>$AREA = 4 * \pi * R^2$<br>$DIAMETER = 2 * R$ |
| trough | $VOLUME\ OF\ CONTENTS = 1/2 * L * WC * AC$<br>$WC * A = AC * W$ |







## KNOWLEDGE ABOUT OBJECTS

| | |
|---|---|
| cone | is a conical object |
| conical | has altitude, contents, base with radius |
| disk | has radius, width, volume, and area |
| sphere | is a spherical object |
| spherical | has area, volume, diameter, and radius |
| trough | has altitude, width, and length |

## TABLE 4: CONTINUED

### KEY WORDS FOR DETERMINING PROBLEM TYPE

DISTANCE KEYS

approaching, distance, separating; ladder, pole, rail; shadow

VOLUME KEYS

altitude, area, diameter, radius, surface, volume

### ADJECTIVE TYPES

conical - shape     cylindrical - shape     green - color
rectangular - shape     red - color     spherical - shape

### DIRECTION WORDS

| | | | | |
|---|---|---|---|---|
| above +K | east +I | horizontally +I | level +I | north -J |
| over +K | south +J | vertically +K | west -I | |

### OTHER KNOWLEDGE

| | |
|---|---|
| altitude | implied by the words deep, depth, height, high, rising, and tall sometimes by surface and level |
| AM | implies the number preceding it in hours |
| balloon | spherical, it and its contents have the same volume, radius, etc. |
| heap | it and its contents have the same volume, etc. |
| Joe | is a person |
| length | is implied by the words long and lengthening |
| level | in a noun phrase implies altitude |
| Mary | is a person |
| midnight | zero hours |
| moving | implies position |
| noon | 12 hours |
| pile | it and its contents have the same volume, etc. |
| PM | adds 12 hours to the number preceding it |
| surface | if followed by AREA implies area, if followed by a verb which indicates altitude (such as rising) implies altitude, else implies area |
| vertical surface | is implied by fence, house, or wall |

This was easy to find since both of these facts were hung from the same node on the structure. This equation is solved for the value of time, which in turn is substituted into the differentiated equation. The latter is then simplified, and the program prints out

```
(THE ANSWER IS)
(TIMES .53132943 IN (EXPT SEC -1.0) (EXPT PI -0.33333332))
```

In conventional notation this would be .53 $(\pi)^{-1/3}$ IN / SEC.

## VII. CONCLUSIONS AND SUGGESTIONS FOR FURTHER WORK

In Chapter 2 we noted several aspects of STUDENT which we felt needed further work. Let us compare CARPS and STUDENT in these areas.

1) By storing information in terms of structures, CARPS is better able to recognize that two phrases describe the same object.

2) Again because of the use of structures CARPS can gather information about an object in piecemeal fashion. STUDENT was essentially required to generate one equation for each sentence in the problem description. In calculus word problems it is not uncommon to have two or three sentences providing information for one equation.

3) CARPS to a limited degree is able to use its knowledge to parse its input sentences. For example we saw in Chapter 5 how ALTITUDE was interpreted as ALTITUDE OF THE FILTER because CARPS knew that since the filter was a cone and cones have altitudes, the filter had an altitude. There was no similar capability in STUDENT.

4) Whereas STUDENT has only one solution method (i.e., solution of linear equations), CARPS has several and can decide which is appropriate for a given problem. CARPS' machinery for solving its equations (e.g., differentiation, simplification) is also more complex than STUDENT's.

5) CARPS utilizes a more sophisticated grammatical analysis of the sentences than STUDENT. This is used both in breaking up sentences and in generating the internal structures. Up to now 14 problems have been solved by CARPS. These are listed in appendix A along with any modifications that were made to the original textbook statement of the

problems in the case that they were taken from a text. We believe that CARPS generally requires less modification of the original problem statement in order to obtain a solution than did STUDENT. Problems 2 and 12 indicate that CARPS is able to handle significantly different wordings of the same problem. Similar variations in the statements of problems would, of course, increase the number of problems solvable by CARPS.

Many of the improvements that we claim for CARPS were necessitated by the increased complexity of the problems that we expected it to solve. However many weaknesses of STUDENT are still present in some form in our design.

1) Probably the most important weakness in the program is due to its dependence on key words to signify the type of problem (i.e, distance or volume) and the method of solution to be used. What one would like to have in a calculus problem solver is a program which would use the information presented in the problem to figure out relationships among the elements (e.g., similar triangles) and actually propose the method of solution. Such a program would probably require a "geometry problem solver", a routine which can be asked questions like "what is a relationship between the radius and the altitude of the cone A?" or "What is the area of a parallelogram with sides a and b and base angles c and d?". The answers which could be provided by such a program are currently built into CARPS' data base for several cases, but this scheme severely limits the power of the program.

2) Another weakness of CARPS is its limited knowledge of English syntax. It would not be too difficult for CARPS to learn new syntactic rules by adding these rules to its CONVERT subroutines. Actually what would be more satisfying would be a different method of parsing the sentences into components of structures. Currently the CONVERT rules are attempted one at a time until one matches the sentence. A better approach would be an incremental left to right parse which, when finished with the sentence, would have translated it into the internal model. Such a routine most likely would switch between several levels of analysis. At one end we would have purely syntactic considerations, and on the other a semantic analysis based upon the information from its general world knowledge. The semantic information gathered in the begining of a sentence could be used to analyze later parts of the same sentence. A scheme such as this is described in Winograd (11).

3) A very powerful calculus word problem solver will require a good deal of "common sense" knowledge. Consider this problem which we gave to CARPS

> (A LADDER 20.0 FT LONG LEANS AGAINST A HOUSE /. FIND THE RATE AT WHICH THE TOP OF THE LADDER IS MOVING DOWNWARD IF ITS FOOT IS 12.0 FT FROM THE HOUSE AND MOVING AWAY AT THE RATE 2.0 FT PER SEC /.)

Much to our surprise CARPS was not able to solve it. A closer look at the problem shows why. The last phrase mentions that the ladder is moving at the rate 2 ft per second. CARPS has, as an internal check, the requirement that associated with each velocity must be the direction of the velocity. The point is that the problem never gave this direction. Most people however, would assume that it was moving directly away from the house. The reason of course, is that a familiarity with ladders or gravity tells us that this is the most likely way for it to be moving.

Nor is this an isolated incident. Consider the problem

> (A BARGE WHOSE DECK IS 10 FT BELOW THE LEVEL OF A DOCK IS BEING DRAWN IN BY MEANS OF A CABLE ATTACHED TO THE DECK AND PASSING THROUGH A RING ON THE DOCK. WHEN THE BARGE IS 24 FT FROM AND APPROACHING THE DOCK AT 3/4 FT / SEC, HOW FAST IS THE CABLE BEING PULLED IN?)

Make a sketch of this situation for yourself. Most all people will draw

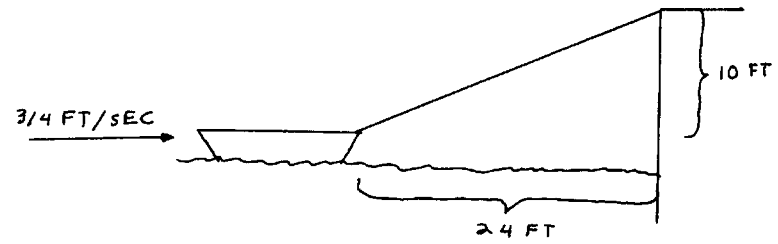


Fig. 7

Clearly when we say APPROACHING THE DOCK we mean at the level of the boat. Once again information of gravity would lead to this result. Yet there are still further difficulties in the problem as stated. The phrase "24 FT FROM ... THE DOCK" also means at the level of the boat. Consider instead the problem

> (A BOY IS FLYING A KITE AT A HEIGHT 150 FT /. IF THE KITE MOVES HORIZONTALLY AWAY FROM THE BOY AT THE RATE 20.0 FT PER SEC /, HOW FAST IS THE STRING BEING PAID OUT WHEN THE KITE IS 250.0 FT FROM HIM ?)

It is just as clear that this time the picture is

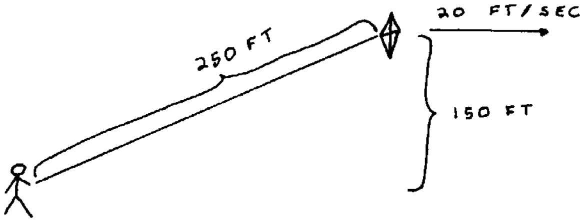


Fig. 8

So the phrase "from him" means the total distance from the kite to the boy. The difference here is that while docks extend downwards (to the level of the boat), kites do not.

Semantic difficulties such as these arise again and again in calculus problems. While we do not present any plan for the manner in which this information should be incorporated, it is clear that a great deal of "real world" knowledge is needed in solving calculus word problems, and for that matter in the understanding of natural language in general.

## APPENDIX A

1 (WATER IS FLOWING INTO A CONICAL FILTER AT THE RATE OF 15.0 CUBIC INCHES PER SECOND /. IF THE RADIUS OF THE BASE OF THE FILTER IS 5.0 INCHES AND THE ALTITUDE IS 10.0 INCHES /, FIND THE RATE AT WHICH THE WATER LEVEL IS RISING WHEN THE VOLUME IS 100.0 CUBIC INCHES /.)

2 (A MAN 6.0 FT TALL WALKS AT THE RATE 5.0 FT PER SECOND TOWARD A STREET LIGHT WHICH IS 16.0 FT ABOVE THE GROUND /. AT WHAT RATE IS THE TIP OF HIS SHADOW MOVING?)

Taken from Thomas (8), page 100. The problem originally asked two questions, only the first is in our problem.

3 (SHIP P IS 15.0 MILES EAST OF O AND MOVING WEST AT 20.0 MILES PER HOUR; SHIP B /, 60.0 MILES SOUTH OF O /, IS MOVING NORTH AT 15.0 MILES PER HOUR /. AT WHAT RATE ARE THEY APPROACHING OR SEPARATING AFTER 1.0 HOUR /.)

Problem taken from Ayres (1), page 59. Three changes were made. SHIP P was originally SHIP A (the A would have been removed along with all THE's), the question originally read "Are they approaching or separating after 1 hr and at what rate?", and there were two other questions.

4 (LADDER 20.0 FEET LONG LEANS AGAINST A HOUSE/. FIND THE RATE AT WHICH THE TOP OF THE LADDER IS MOVING IF ITS FOOT IS 12.0 FEET FROM THE HOUSE AND MOVING AWAY FROM THE HOUSE AT THE RATE 2.0 FEET PER SECOND/.)

Taken from Ayres page 59. The problem originally asked two questions.

5 (A TRAIN STARTING AT 11.0 AM /, TRAVELS EAST AT 45.0 MILES PER HOUR WHILE ANOTHER /, STARTING AT NOON FROM THE SAME POINT /, TRAVELS SOUTH AT 60.0 MILES PER HOUR /. HOW FAST ARE THEY SEPARATING AT 3.0 PM ?)

Taken from Ayres page 59, no changes.

6 (GAS IS ESCAPING FROM A SPHERICAL BALLOON AT THE RATE OF 2.0 CUBIC FEET PER MINUTE /. HOW FAST IS THE SURFACE AREA SHRINKING WHEN THE RADIUS IS 12.0 FEET ?)

Taken from Ayres page 57, no changes.

7 (A BALLOON IS RISING VERTICALLY OVER A POINT B AT THE RATE 15.0 FEET PER SECOND /. A POINT C IS LEVEL WITH B AND IS 30.0 FEET FROM B /. WHEN THE BALLOON IS 40.0 FEET FROM B /, AT WHAT RATE IS ITS DISTANCE FROM C CHANGING ?)

8 (A SHIP IS 30.0 MILES SOUTH OF POINT O AND TRAVELING EAST AT 25.0 MILES PER HOUR /. HOW FAST IS THE DISTANCE FROM THE SHIP TO O INCREASING ?)

9 (UPON BEING HEATED /, A METAL DISK EXPANDS /. THE RADIUS OF THE DISK LENGTHENS AT THE RATE OF .01 IN PER SEC /. CALCULATE THE RATE AT WHICH THE AREA OF THE DISK IS INCREASING WHEN THE RADIUS IS 3.0 IN /.)

Taken from Lightstone (5) page 145. The problem originally had two questions and the first two sentences above were connected by the phrase "in such a manner that".

10 (SAND FALLS ONTO A CONICAL PILE AT THE RATE OF 10.0 FT3 PER MIN /. THE RADIUS OF THE BASE OF THE PILE IS ALWAYS EQUAL TO ONE HALF OF ITS ALTITUDE /. HOW FAST IS THE ALTITUDE OF THE PILE INCREASING WHEN IT IS 5 FT DEEP ?)

Taken from Thomas page 106, no changes.

11 (A RECTANGULAR TROUGH IS 8.0 FT LONG /, 2.0 FT WIDE /, AND 4.0 FT DEEP /. IF WATER FLOWS INTO THE TROUGH AT THE RATE 2.0 FT3 PER MIN /, HOW FAST IS THE SURFACE OF THE WATER RISING WHEN THE WATER IS 1.0 FT DEEP ?)

Taken from Ayres page 59. Rather than WIDE the problem had "across the top", instead of INTO THE TROUGH the problem had "in", and instead of SURFACE OF THE WATER the problem had just "surface".

12 (A MAN WALKS AWAY FROM A STREET LIGHT AT 4.0 FT PER SECOND /. HOW FAST IS THE POSITION OF THE TIP OF HIS SHADOW CHANGING IF HE IS 5.0 FEET TALL /, WHILE IT IS 20.0 FEET TALL /.)

[illegible] A STREET LIGHT WHOSE HEIGHT IS 30.0 FEET /. IF JOE [illegible] TRAVELING AT 10.0 MILES PER HOUR /, FIND THE [illegible] IS LENGTHENING ?)

[illegible] SPHERE IS INCREASING AT THE RATE OF 4.0 SQUARE [illegible] WHAT RATE IS THE DIAMETER CHANGING AFTER

Bibliography

1 Ayres, F., Theory and Problems of Differential and Integral Calculus, Schaum Publishing Co., New York, 1964.

2 Bobrow, D. G. "Natural Language Input for a Computer Problem Solving System", Report MAC-TR-1, Project MAC, M.I.T., Cambridge, Mass., June 1964, Reprinted in Semantic Information Processing, MIT Press, 1968.

3 Guzman A., and McIntosh, H. V., "A Miscellaney of CONVERT Programming", Memorandum MAC-M-346, Project MAC, M.I.T., Cambridge, Mass., April 1967.

4 Guzman A., and McIntosh, H. V., "CONVERT", Communications of the ACM, Vol. 9, No. 8, August 1966.

5 Lightstone, A. H., Concepts of Calculus, Harper and Row, New York, 1965.

6 Moses, J., "Symbolic Integration", Report MAC-TR-47, Project MAC, M.I.T., Cambridge, Mass., December 1967.

7 Raphael, B., "Operation of A Semantic Question-Answering System" Artificial Intelligence Project - RLE and M.I.T. Computation Center, Memo 59, November 1963.

8 Thomas, G. B., Calculus and Analytic Geometry, Addison Wesley Publishing Co., Reading, Mass., 1959.

9 Weizenbaum, J., "ELIZA-A Computer Program For The Study of Natural Language Communication Between Man and Machine", Communications of the ACM, Vol. 9,No. 1, January 1966, pp 36-43.

10 Weizenbaum, J., "Contextual Understanding by Computers", Communications of the ACM, Vol. 10, No. 8, August 1967, pp 474-480.

11 Winograd, T. "An Interpretive Theory of Language" Unpublished term paper, 18.16J, Department of Mathematics, April 1968.

UNCLASSIFIED
Security Classification

## DOCUMENT CONTROL DATA - R&D

*(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)*

| | |
|---|---|
| 1. ORIGINATING ACTIVITY *(Corporate author)*<br>Massachusetts Institute of Technology<br>Project MAC | 2a. REPORT SECURITY CLASSIFICATION<br>UNCLASSIFIED<br>2b. GROUP<br>None |

3. REPORT TITLE

CARPS, A Program Which Solves Calculus Word Problems

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

Masters Thesis, Department of Electrical Engineering, July 1968

5. AUTHOR(S) *(Last name, first name, initial)*

Charniak, Eugene

| | | |
|---|---|---|
| 6. REPORT DATE<br>July 1968 | 7a. TOTAL NO. OF PAGES<br>64 | 7b. NO. OF REFS<br>11 |

| | |
|---|---|
| 8a. CONTRACT OR GRANT NO.<br>Office of Naval Research, Nonr-4102(01)<br>b. PROJECT NO.<br>NR 048-189<br>c.<br>RR 003-09-01<br>d. | 9a. ORIGINATOR'S REPORT NUMBER(S)<br>MAC-TR-51<br>9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)* |

10. AVAILABILITY/LIMITATION NOTICES

Distribution of this document is unlimited.

| | |
|---|---|
| 11. SUPPLEMENTARY NOTES<br>None | 12. SPONSORING MILITARY ACTIVITY<br>Advanced Research Projects Agency<br>3D-200 Pentagon<br>Washington, D.C. 20301 |

13. ABSTRACT

CARPS (CAlculus Rate Problem Solver) is a program written to solve calculus word problems, but is restricted to rate problems. The overall plan of the program is similar to Bobrow's STUDENT, the primary difference being the introduction of "structures" as the internal model in CARPS. Structures are stored internally as trees. Each structure is designed to hold the information gathered about one object.

A description of CARPS is given by working through two problems, one in great detail. Also included is a critical analysis of STUDENT.

14. KEY WORDS

Calculus problem solving
Computers
Machine-aided cognition
Multiple-access computers
Natural-language programs
On-line computers
Real-time computers
Time-sharing
Time-shared computers

DD FORM 1 NOV 65 1473 (M.I.T.)

UNCLASSIFIED
Security Classification