8

A SEMANTIC ASSOCIATIONAL MEMORY NET THAT LEARNS AND ANSWERS QUESTIONS

(SAMENLAQ)

by

Stuart C. Shapiro, G. H. Woodmansee,
Myron W. Krueger

Computer Sciences Technical Report #8

January 1968

TABLE OF CONTENTS

ABSTRACT

This paper describes a general semantic memory structure and associated executive functions, which, using the memory, are capable of learning and answering questions. The system is capable of three types of learning: by being explicitly told facts; by deducing facts implied by a number of previously stored facts; by induction from a given set of examples. The memory structure is a net built up of binary relations, a relation being a label on the edge joining two nodes. The relations, however, are also nodes and so can be related to other nodes which may also be used as relations. Most significantly, a relation can be related to one or more alternate definitions in terms of compositions of other relations and restrictions on intermediate nodes. Throughout this work an attempt was made to maintain complete generality and thus allow the system to be used in a wide variety of applications without change. In fact, it could be used for several different purposes simultaneously. The system, as described, is presently programmed in SNOBOL, a string manipulation language, and is running at the University of Wisconsin on a C.D.C. 3600 computer.

1.0 INTRODUCTION

We have been examining the memory structure and search procedures required by a general question answering program; one which is capable of learning and reorganizing itself based on its experience. Ideally, such a program would accept questions posed in natural language. However, since our interests lie mainly in the memory structure itself, we have bypassed the input parsing and output formatting problems for the present. Thus, our immediate goal has been to develop a general memory structure and to explore various techniques for growing, reorganizing, and interrogating it.

The memory structure we agreed upon is a semantic, associative net. It is semantic in the sense that a word is defined by its relations to other words, these relations being learned through general experience rather than by reading a dictionary (which is the way Quillian's "semantic memory" [6] learns). This type of memory structure appears to be general enough to accommodate a wide variety of contexts simultaneously. While many of the examples presented here are drawn from the domain of family relations, neither the memory structure nor the search procedures are in any way dependent upon that context.

The current configuration of the program contains a set of integrated functions which provides for the entering and retrieving of explicit information as well as discovering and assimilating implicit information.

1.1 RELATED PREVIOUS WORK

Several systems that are related to the one described here have been presented. We shall briefly mention these by Green et. al. [3], Lindsay [4], Levien and Maron [5], Elliott [1], Raphael [7], and Quillian [6].

Green was primarily interested in the syntactic problems involved in answering questions posed in natural English. His system answers questions about baseball, so its data is organized in an outline structure which explicitly provides six pieces of information for each game: the month, place, day of the month, game serial number, first team and score, second team and score. Lindsay's system deals with family relations, and its data is organized very much like a genealogy chart or family tree, the basic structure being the family unit with pointers to the husband, wife, offspring, husband's parents, and wife's parents. Levien and Maron's basic storage is much more general than these, being in coded relational statements of the form x_1Rx_2. Their input, however, is very much dependent on their problem area, literature search, and consists of a series of involved forms that must be filled in manually. Raphael also has a general structure and makes a start toward using general relations. Each relation he uses, however, requires specific programs to deal with it, and to add new relations one must write the necessary program functions. Elliott expands on this by having new relations defined in terms of nine properties. Because certain combinations of these properties cannot co-occur, they form thirty-two classes of relations. Each class has special sets of functions to deal with any relation in the class. He also allows relations to be defined in terms of unrestricted logical statements using previously defined relations, but these can only be used in answering questions, not in storing facts. Quillian uses the most general structure of those mentioned, with words represented by type nodes which point to their dictionary definitions in terms of token nodes which point to the type nodes of the words they represent which, in turn, point to their definitions. The major defect of this work when compared to the others we have mentioned is that the only way definitions can be added to the structure is by being hand coded in their entirety. The most general memory structures heretofore described have been the cognitive structures of Reitman [8], where all entities are lists described by attribute-value sublists, with no relation or entity having a special status due to specific programming. This is the most immediate intellectual ancestor of the work described in the rest of this paper.

1.2 ACKNOWLEDGMENTS

The authors gratefully acknowledge the guidance of Professor Leonard Uhr of the University of Wisconsin Computer Sciences Department throughout the course of this work and the preparation of this paper, and the valuable comments and criticisms of Professor Larry Travis, also of the University of Wisconsin Computer Sciences Department.

2.0 A SAMPLE RUN

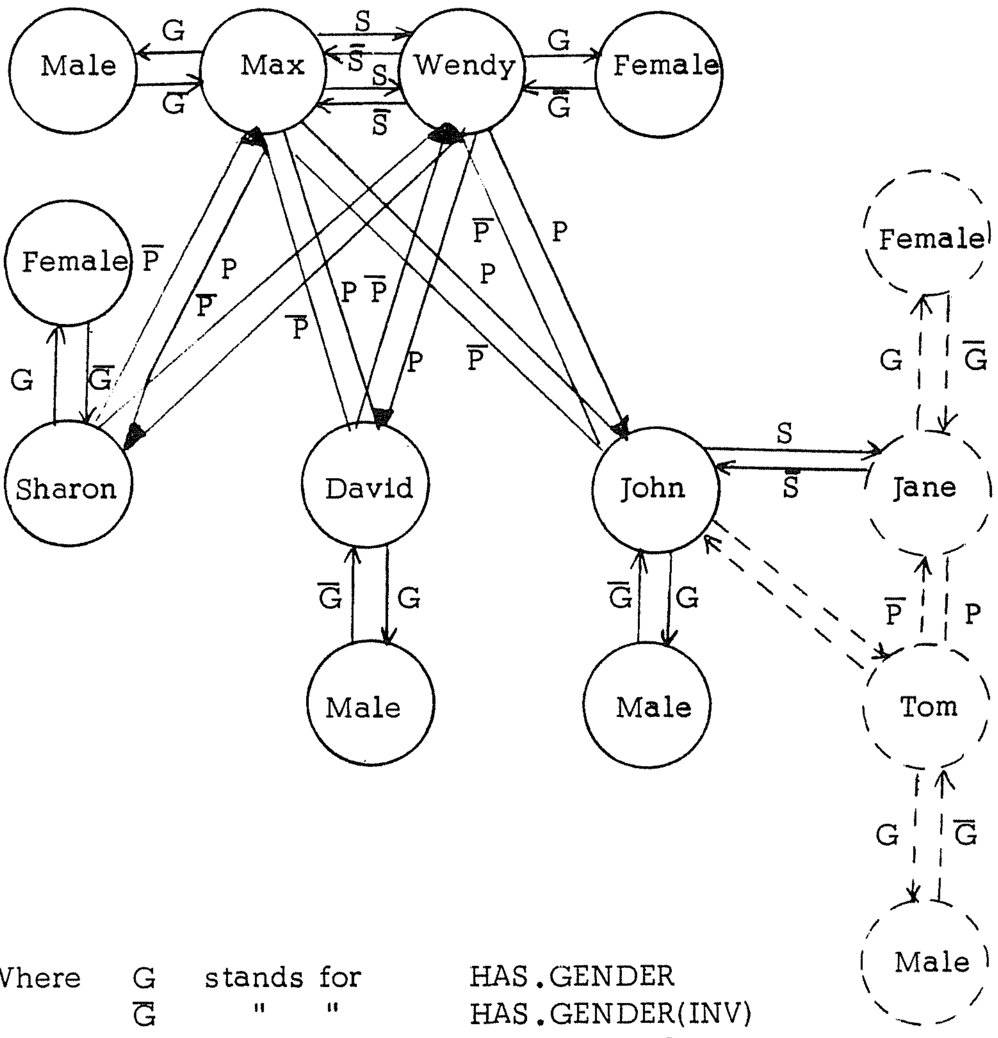
Prior to beginning a detailed description of the program, the reader may find it helpful to follow through the sample run described in this section. Not only will this give him a feeling for the program's capabilities, but it will provide him with concrete examples of program features that will be described later in more general terms. The run deals with family-relations, a domain familiar to all. In the process of following the sample, the reader is asked to keep in mind that the memory structure in no way depends upon the problem domain of family-relations.

The program classifies each input string as being either a statement or a question. The string is then prefixed by its classification and printed out.

```
STATEMENT - IS.SPOUSE.OF MEMBER FAMILY.RELATION
STATEMENT - IS.SPOUSE.OF(INV) MEMBER FAMILY.RELATION
STATEMENT - IS.PARENT.OF MEMBER FAMILY.RELATION
STATEMENT - IS.PARENT.OF(INV) MEMBER FAMILY.RELATION
STATEMENT - MAX HAS.GENDER MALE
STATEMENT - MAX IS.SPOUSE.OF WENDY
STATEMENT - MAX IS.PARENT.OF JOHN
STATEMENT - MAX IS.PARENT.OF DAVID
STATEMENT - MAX IS.PARENT.OF SHARON
STATEMENT - WENDY HAS.GENDER FEMALE
STATEMENT - WENDY IS.SPOUSE.OF MAX
STATEMENT - WENDY IS.PARENT.OF JOHN
STATEMENT - WENDY IS.PARENT.OF DAVID
STATEMENT - WENDY IS.PARENT.OF SHARON
STATEMENT - JOHN HAS.GENDER MALE
STATEMENT - DAVID HAS.GENDER MALE
STATEMENT - SHARON HAS.GENDER FEMALE
```

At this point, the program has processed the above inputs. The structure represented in Figure 1 by solid lines indicates the program's current knowledge of the group's interrelations.

Figure 1



Where		stands for	
	G	stands for	HAS.GENDER
	$\bar{G}$	" "	HAS.GENDER(INV)
	S	" "	IS.SPOUSE.OF
	$\bar{S}$	" "	IS.SPOUSE.OF(INV)
	P	" "	IS.PARENT.OF
	$\bar{P}$	" "	IS.PARENT.OF(INV)

*Note that while Figure 1 correctly indicates the relations among the various people in the net, it is not an accurate representation of the actual internal memory configuration. Appendix B describes how an accurate graphical representation may be constructed. In many cases however, partially complete representations such as Figure 1 are adequate to describe the salient features of the structure being considered.

We are now in a position to ask several questions based upon the current memory net structure.

```
 QUFSTION - MAX IS.PARFNT.OF JOHN
ANSWFR -     TRUF


 QUFSTION - WFNDY IS.PARFNT.OF MAX
ANSWER -     FALSE
```

These questions represent the simplest form of memory interrogation. The next question type requires that the program recognize appropriate interrogative words -- in this particular case ... "WHO." We thus inform the program that "WHO" is a question word.

```
STATFMFNT - WHO IS QUFSTION
```

We may now ask questions of a slightly more general nature.

```
 QUFSTION - WHO HAS.GFNDFR MALF
ANSWFR - MAX
   AND   JOHN
   AND   DAVID


 QUFSTION - WFNDY IS.PARFNT.OF WHO
ANSWFR - JOHN
   AND   DAVID
   AND   SHARON


 QUESTION - WHO IS.SPOUSF.OF/IS.PARFNT.OF DAVID
ANSWFR - WFNDY
   AND   MAX
```

The last of these requires that the program deal with relation composition. Note that in answering the first question the program starts at the "MALE" node and works towards the answer nodes "MAX", "JOHN", and "DAVID" via the "GENDER(INV)" relation.

Information can be added to the net at any time. This is accomplished by simply supplying additional input statements.

```
STATEMENT - JOHN IS.SPOUSE.OF JANE
STATEMENT - JANE HAS.GENDER FEMALE
STATEMENT - JOHN IS.PARENT.OF TOM
STATEMENT - JANE IS.PARENT.OF TOM
STATEMENT - TOM HAS.GENDER MALE
```

These five statements modify the previous memory structure. The new structure is shown in figure #1 by both solid and dashed lines.

The program's entire store of information concerning family-relations has been supplied by the previous twenty-two input statements. It has seen the relations IS.SPOUSE.OF, IS.PARENT.OF, HAS.GENDER and their converses IS.SPOUSE.OF(INV), IS.PARENT.OF(INV), HAS.GENDER(INV). It has not seen IS.SON.OF or IS.SON.OF(INV).

```
QUESTION - WHO IS.SON.OF JOHN
```

Note that to answer this question, the program must work from "JOHN" via the "IS.SON.OF(INV)" relation to an answer. However the program has never seen the relation IS.SON.OF(INV) and thus responds:

```
 WITH THE INFORMATION I NOW HAVE I CANNOT REDEFINE IS.SON.OF(INV)
SO THAT IT CAN BE APPLIED TO JOHN.
IF YOU WISH TO ACCEPT THIS AS YOUR ANSWER ENTER - OK
IF YOU WISH TO SUPPLY AN ANSWER, ENTER A STATEMENT OF THE FORM -
 JOHN IS.SON.OF(INV) X     OR     IS.SON.OF(INV) EQUIV X
OTHERWISE ENTER A SERIES OF PAIRS X Y SUCH THAT X IS.SON.OF(INV) Y
FOLLOWED BY THE CLASS OF RELATIONS IS.SON.OF(INV) BELONGS TO
FOLLOWED BY AN INTEGER INDICATING THE MAXIMUM NUMBER OF RELATIONS
THAT MAY BE COMPOSED TO GIVE AN ALTERNATE DEFINITION OF IS.SON.OF(INV).
I WILL THEN ATTEMPT TO FIND A MORE SATISFACTORY  ANSWER TO YOUR QUESTION.
```

We elect to follow the latter course.

```
 EXAMPLES - MAX     JOHN
            WENDY    JOHN
            MAX     DAVID
CLASS - FAMILY.RELATION
DEPTH - 3
```

The program responds:

```
THANK YOU.
```

and retires briefly to reflect upon the problem.

```
 FROM THE EXAMPLES YOU HAVE GIVEN ME, I WOULD GUESS THAT
IS.SON.OF(INV) IS THE SAME AS
     IS.PARENT.OF/(HAS.GENDER-MALE,)
 OR  IS.SPOUSE.OF(INV)/IS.PARENT.OF/(HAS.GENDER-MALE,)
 OR  IS.SPOUSE.OF/IS.PARENT.OF/(HAS.GENDER-MALE,)
 OR  IS.PARENT.OF/IS.PARENT.OF(INV)/IS.PARENT.OF/(HAS.GENDER-MALE,)
 I CAN NOT APPLY IS.SON.OF(INV) TO JOHN
BUT I KNOW IS.SON.OF(INV) IS THE SAME AS
    IS.PARENT.OF/(HAS.GENDER-MALE,)
 OR IS.SPOUSE.OF(INV)/IS.PARENT.OF/(HAS.GENDER-MALE,)
 OR IS.SPOUSE.OF/IS.PARENT.OF/(HAS.GENDER-MALE,)
 OR IS.PARENT.OF/IS.PARENT.OF(INV)/IS.PARENT.OF/(HAS.GENDER-MALE,)
```

Although the program "knows" that JOHN IS.SPOUSE.OF JANE, it does not know that "IS.SPOUSE.OF" is symmetric and thus can not solve the problem since in order to do so it must know that JOHN IS.SPOUSE.OF (INV) JANE.

```
 WITH THF INFORMATION I NOW HAVF I CANNOT REDEFINE IS.SPOUSE.OF(INV)
SO THAT IT CAN BE APPLIED TO JOHN.
IF YOU WISH TO ACCEPT THIS AS YOUR ANSWER ENTER - OK
IF YOU WISH TO SUPPLY AN ANSWER, ENTER A STATEMENT OF THE FORM -
 JOHN IS.SPOUSE.OF(INV) X      OR      IS.SPOUSE.OF(INV) EQUIV X
OTHERWISE ENTER A SERIES OF PAIRS X Y SUCH THAT X IS.SPOUSE.OF(INV) Y
FOLLOWED BY THE CLASS OF RELATIONS IS.SPOUSE.OF(INV) BELONGS TO
FOLLOWED BY AN INTEGER INDICATING THE MAXIMUM NUMBER OF RELATIONS
THAT MAY BE COMPOSED TO GIVE AN ALTERNATE DEFINITION OF IS.SPOUSE.OF(INV).
I WILL THEN ATTEMPT TO FIND A MORE SATISFACTORY  ANSWER TO YOUR QUESTION.
```

Since the program has done as well as could be expected with the current information store, we give it a hint.

```
STATEMENT - IS.SPOUSE.OF(INV) EQUIV IS.SPOUSE.OF/

THANK YOU.
```

The program now knows that IS.SPOUSE.OF(INV) is equivalent to IS.SPOUSE.OF. eg. "IS.SPOUSE.OF" is a symmetric relation. This is all the information it needs to solve the problem and it quickly responds:

```
 I CAN NOT APPLY IS.SPOUSE.OF(INV) TO JOHN
BUT I KNOW IS.SPOUSE.OF(INV) IS THE SAME AS
    IS.SPOUSE.OF/
 I FIGURE JOHN IS.SPOUSE.OF(INV) JANE,
 I FIGURE JOHN IS.SON.OF(INV) TOM,
ANSWER - TOM
```

which is correct as the dubious reader may quickly verify.

In the process of answering the original question, the program has discovered an alternate definition of IS.SON.OF(INV) in terms of relations it already knew. With the addition of the general information

```
STATEMFNT - IS.SON.OF MFMRFR FAMILY.RFLATION
STATFMENT - IS.SON.OF(INV) MFMRFR FAMILY.RFLATION
```

it is in a position to answer more specific questions concerning these two new relations.

```
QUESTION - WHO IS.SON.OF MAX
```

This time the program has all necessary information and responds after some introspection -

```
 I CAN NOT APPLY IS.SON.OF(INV) TO MAX
BUT I KNOW IS.SON.OF(INV) IS THF SAMF AS
    IS.PARENT.OF/(HAS.GENDER-MALE,)
 OR IS.SPOUSE.OF(INV)/IS.PARFNT.OF/(HAS.GFNDER-MALF,)
 OR IS.SPOUSE.OF/IS.PARENT.OF/(HAS.GENDFR-MALF,)
 OR IS.PARENT.OF/IS.PARENT.OF(INV)/IS.PARENT.OF/(HAS.GENDER-MALE,)
 I FIGURE MAX IS.SON.OF(INV) JOHN,DAVID,
ANSWFR - JOHN
   AND   DAVID
```

The reader should now be able to follow the program output directly. The following examples further indicate the program's ability to interrogate its environment when necessary as well as use equivalent definitions of relations in order to answer questions about nodes not explicitly related by those relations.

```
QUFSTION - WHO IS.SON.OF WFNDY
```

```
 I CAN NOT APPLY IS.SON.OF(INV) TO WENDY
BUT I KNOW IS.SON.OF(INV) IS THE SAME AS
    IS.PARENT.OF/(HAS.GENDER-MALE,)
 OR IS.SPOUSE.OF(INV)/IS.PARENT.OF/(HAS.GENDER-MALE,)
 OR IS.SPOUSE.OF/IS.PARENT.OF/(HAS.GENDER-MALE,)
 OR IS.PARENT.OF/IS.PARENT.OF(INV)/IS.PARENT.OF/(HAS.GENDER-MALE,)

 I FIGURE WENDY IS.SON.OF(INV) JOHN,DAVID,
ANSWER - JOHN
   AND   DAVID


STATEMENT - HAS.SON MEMBER FAMILY.RELATION
STATEMENT - HAS.SON(INV) MEMBER FAMILY.RELATION

 QUESTION - JOHN HAS.SON WHO

 WITH THE INFORMATION I NOW HAVE I CANNOT REDEFINE HAS.SON
SO THAT IT CAN BE APPLIED TO JOHN.
IF YOU WISH TO ACCEPT THIS AS YOUR ANSWER ENTER - OK
IF YOU WISH TO SUPPLY AN ANSWER, ENTER A STATEMENT OF THE FORM -
 JOHN HAS.SON X    OR     HAS.SON EQUIV X
OTHERWISE ENTER A SERIES OF PAIRS X Y SUCH THAT X HAS.SON Y
FOLLOWED BY THE CLASS OF RELATIONS HAS.SON BELONGS TO
FOLLOWED BY AN INTEGER INDICATING THE MAXIMUM NUMBER OF RELATIONS
THAT MAY BE COMPOSED TO GIVE AN ALTERNATE DEFINITION OF HAS.SON.
I WILL THEN ATTEMPT TO FIND A MORE SATISFACTORY  ANSWER TO YOUR QUESTION.

 EXAMPLES - MAX    JOHN
           WENDY    DAVID
           MAX    DAVID
CLASS - FAMILY.RELATION
DEPTH - 3

 THANK YOU.

 FROM THE EXAMPLES YOU HAVE GIVEN ME, I WOULD GUESS THAT
HAS.SON IS THE SAME AS
     IS.SON.OF(INV)/(HAS.GENDER-MALE,)
 OR  IS.PARENT.OF/(HAS.GENDER-MALE,)
 OR  IS.SPOUSE.OF(INV)/IS.SON.OF(INV)/(HAS.GENDER-MALE,)
 OR  IS.SPOUSE.OF(INV)/IS.PARENT.OF/(HAS.GENDER-MALE,)
 OR  IS.SPOUSE.OF/IS.SON.OF(INV)/(HAS.GENDER-MALE,)
 OR  IS.SPOUSE.OF/IS.PARENT.OF/(HAS.GENDER-MALE,)
 OR  IS.SON.OF(INV)/IS.PARENT.OF(INV)/IS.SON.OF(INV)/(HAS.GENDER-MALE,)
 OR  IS.SON.OF(INV)/IS.PARENT.OF(INV)/IS.PARENT.OF/(HAS.GENDER-MALE,)
 OR  IS.PARENT.OF/IS.PARENT.OF(INV)/IS.SON.OF(INV)/(HAS.GENDER-MALE,)
 OR  IS.PARENT.OF/IS.PARENT.OF(INV)/IS.PARENT.OF/(HAS.GENDER-MALE,)
 OR  IS.PARENT.OF/IS.PARENT.OF(INV)/IS.SON.OF(INV)/(HAS.GENDER-MALE,)
 I CAN NOT APPLY HAS.SON TO JOHN
BUT I KNOW HAS.SON IS THE SAME AS
    IS.SON.OF(INV)/(HAS.GENDER-MALE,)
 OR IS.PARENT.OF/(HAS.GENDER-MALE,)
 OR IS.SPOUSE.OF(INV)/IS.SON.OF(INV)/(HAS.GENDER-MALE,)
 OR IS.SPOUSE.OF(INV)/IS.PARENT.OF/(HAS.GENDER-MALE,)
 OR IS.SPOUSE.OF/IS.SON.OF(INV)/(HAS.GENDER-MALE,)
 OR IS.SPOUSE.OF/IS.PARENT.OF/(HAS.GENDER-MALE,)
 OR IS.SON.OF(INV)/IS.PARENT.OF(INV)/IS.SON.OF(INV)/(HAS.GENDER-MALE,)
 OR IS.SON.OF(INV)/IS.PARENT.OF(INV)/IS.PARENT.OF/(HAS.GENDER-MALE,)
 OR IS.PARENT.OF/IS.PARENT.OF(INV)/IS.SON.OF(INV)/(HAS.GENDER-MALE,)
 OR IS.PARENT.OF/IS.PARENT.OF(INV)/IS.PARENT.OF/(HAS.GENDER-MALE,)

 I CAN NOT APPLY IS.SON.OF(INV) TO JANE
BUT I KNOW IS.SON.OF(INV) IS THE SAME AS
    IS.PARENT.OF/(HAS.GENDER-MALE,)
 OR IS.SPOUSE.OF(INV)/IS.PARENT.OF/(HAS.GENDER-MALE,)
 OR IS.SPOUSE.OF/IS.PARENT.OF/(HAS.GENDER-MALE,)
 OR IS.PARENT.OF/IS.PARENT.OF(INV)/IS.PARENT.OF/(HAS.GENDER-MALE,)

 WITH THE INFORMATION I NOW HAVE I CANNOT REDEFINE IS.SPOUSE.OF
SO THAT IT CAN BE APPLIED TO JANE.
IF YOU WISH TO ACCEPT THIS AS YOUR ANSWER ENTER - OK
IF YOU WISH TO SUPPLY AN ANSWER, ENTER A STATEMENT OF THE FORM -
 JANE IS.SPOUSE.OF X     OR     IS.SPOUSE.OF EQUIV X
OTHERWISE ENTER A SERIES OF PAIRS X Y SUCH THAT X IS.SPOUSE.OF Y
FOLLOWED BY THE CLASS OF RELATIONS IS.SPOUSE.OF BELONGS TO
FOLLOWED BY AN INTEGER INDICATING THE MAXIMUM NUMBER OF RELATIONS
THAT MAY BE COMPOSED TO GIVE AN ALTERNATE DEFINITION OF IS.SPOUSE.OF.
I WILL THEN ATTEMPT TO FIND A MORE SATISFACTORY  ANSWER TO YOUR QUESTION.

 STATEMENT - IS.SPOUSE.OF EQUIV IS.SPOUSE.OF(INV)/

THANK YOU.

 I CAN NOT APPLY IS.SPOUSE.OF TO JANE
BUT I KNOW IS.SPOUSE.OF IS THE SAME AS
    IS.SPOUSE.OF(INV)/

 I FIGURE JANE IS.SPOUSE.OF JOHN,

 I FIGURE JANE IS.SON.OF(INV) TOM,

 I FIGURE JOHN HAS.SON TOM,

ANSWER - TOM


 QUESTION - JANE HAS.SON WHO
 I CAN NOT APPLY HAS.SON TO JANE
BUT I KNOW HAS.SON IS THE SAME AS
    IS.SON.OF(INV)/(HAS.GENDER-MALE,)
 OR IS.PARENT.OF/(HAS.GENDER-MALE,)
 OR IS.SPOUSE.OF(INV)/IS.SON.OF(INV)/(HAS.GENDER-MALE,)
 OR IS.SPOUSE.OF(INV)/IS.PARENT.OF/(HAS.GENDER-MALE,)
 OR IS.SPOUSE.OF/IS.SON.OF(INV)/(HAS.GENDER-MALE,)
 OR IS.SPOUSE.OF/IS.PARENT.OF/(HAS.GENDER-MALE,)
 OR IS.SON.OF(INV)/IS.PARENT.OF(INV)/IS.SON.OF(INV)/(HAS.GENDER-MALE,)
 OR IS.SON.OF(INV)/IS.PARENT.OF(INV)/IS.PARENT.OF/(HAS.GENDER-MALE,)
 OR IS.PARENT.OF/IS.PARENT.OF(INV)/IS.SON.OF(INV)/(HAS.GENDER-MALE,)
 OR IS.PARENT.OF/IS.PARENT.OF(INV)/IS.PARENT.OF/(HAS.GENDER-MALE,)
```

```
I FIGURF JANF HAS.SON TOM,
ANSWFR - TOM
```

This sample run has demonstrated some of the capabilities of SAMENLAQ. The problem domain has been that of family-relations. As mentioned at the beginning of this section, family-relations were chosen for their familiarity and in no way restrict the generality of the program. The general structure of the program is described in succeeding sections and the program's wide domain of application should readily become apparent to the reader. For those desiring further examples, appendix D contains a more general sample run.

3.0 THE PROGRAM

The program consists of three functionally interdependent parts: the memory structure, the learning functions, and the question answering executive.

The memory structure is described in 3.1 and is the repository for all information available to the question answering executive.

During the answering of certain classes of questions, the question answering executive may generate intermediate results which represent new relations between entities already in the memory structure. These relations are learned. In addition to this type of learning, the program is capable of assimilating explicitly stated information. Examples of the various learning mechanisms are given in 3.2.

The question answerer attempts to answer questions directly from information represented explicitly in the net. Failing to accomplish this, it proceeds to call itself recursively. In the process, it generates a hierarchy of goals and subgoals whose satisfaction will result in an answer. This process is described in detail in 3.3.

3.1 INTERNAL REPRESENTATION

All statements and questions are assumed to be in the following form:

NAME1 RELATION NAME2 KEY

where NAME1 and NAME2 correspond to nodes in the memory net, RELATION corresponds to a descriptive link, and blanks serve as name delimeters. KEY is used during question answering and will be explained later. Appendices A and B give a formal definition of the memory structure* and the terms we will be using in describing it. The most important structure in the net is the <u>paren</u> <u>list</u> associated with each node.

Example of a <u>paren</u> <u>list</u>

DOGS :: = (EAT-/1) (LIKE-/2) (IS.MEM-/3) (HAVE-/4)

/1 :: = MEAT,

/2 :: = HUMANS,

/3 :: = ANIMALS,

/4 :: = LEGS, FUR, PAWS,

The <u>paren</u> <u>list</u> represents the links to nodes relevant to the given node and describes exactly how it is related to each of these nodes. The other type of list is called a <u>comma</u> <u>list</u> and contains the names of other nodes associated with the given node in the particular way defined by the relations on the <u>paren</u> <u>list</u>. The <u>paren</u> <u>pair</u> (HAVE-/4) indicates that all dogs have each

*Note that while the BNF memory description given in Appendix A is invariant, the instantaneous configuration of memory comprised of a specific set of names and relations is not. Throughout this paper both the BNF description of memory and the instantaneous memory configuration are referred to as "structure". In each case the context should make clear which aspect of memory is being referenced.

of the elements found on the comma list named /4. For economy, a comma list may contain another slash name as an element. The structure of these examples is represented schematically below in figure #2.

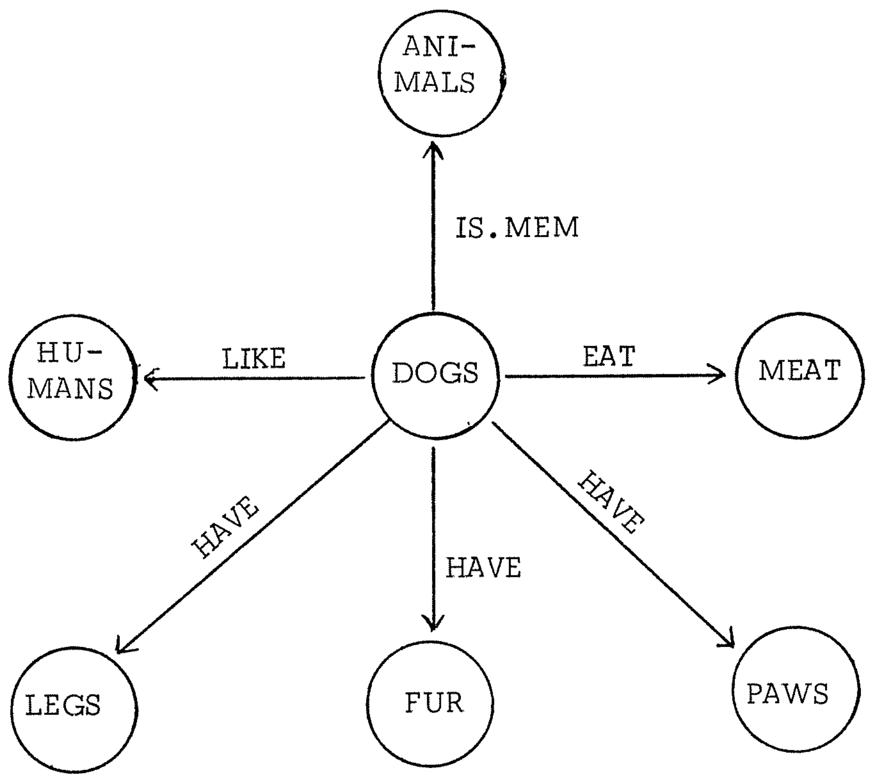


Figure 2

The remaining type of structure is a special type of comma list called an equiv list.

Example of an equiv list:

Given HAS.GRANDFATHER:: = (EQUIV-/6) (IS.MEM-/7)

the equiv list associated with HAS.GRANDFATHER

/6:: = (IS.MEM-HUMAN) HAS.PARENT/HAS.FATHER

(IS.MEM-HUMAN) HAS.PARENT/HAS.PARENT/(GENDER-MALE),

Each element on an equiv list is an alternate definition of the given relation. There are three parts to the alternate definition:

1. a set of restrictions which the first node must satisfy, eg. "(IS.MEM-HUMAN)";
2. a super relation which is the composition of several relations, eg. "HAS.PARENT/HAS.FATHER/"
3. a set of restrictions which the second node must satisfy, eg. "(GENDER-MALE)".

The restrictions are in the form of paren pairs eg. "IS.MEM-HUMAN)" and "(GENDER-MALE)" for which the object nodes "HUMAN" and "MALE" have been explicitly stated. The requirements here are that the first node be a member of the human race and that the second node have male gender.

3.2 LEARNING

The program is currently capable of three functionally different types of memory modification. These are the assimilation of

1. explicitly stated relations between names,
2. relations between names implicitly contained in a question and
3. relational definitions implicitly contained in the net.

It is felt that these mechanisms constitute a powerful means of expanding and reorganizing the memory structure. Their specific workings are described in the succeeding sections.

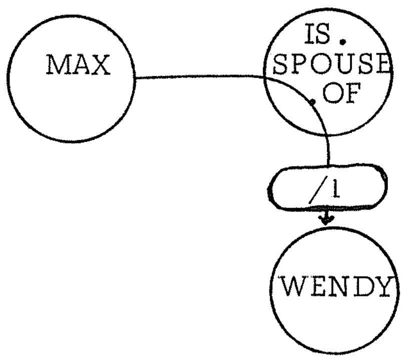
3.2.1 INPUT AND LEARNING

All input to the memory structure is in the format:

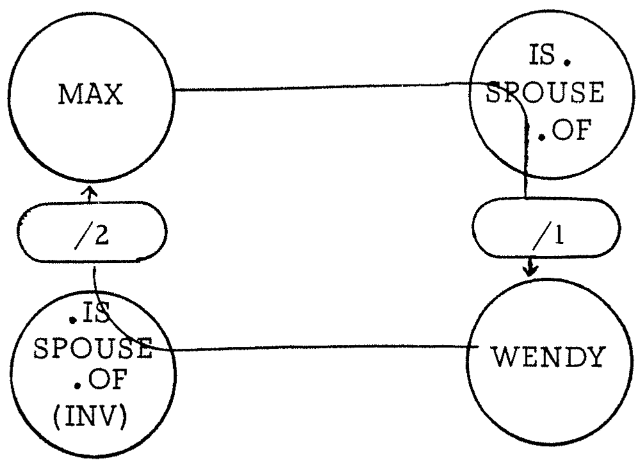
NAME1 RELATION NAME2 KEY

where NAME1 and NAME2 are the names of nodes, and the KEY indicates whether the input is a question (a key of "***" indicates that the input is a question and causes the question answering routines to be called). If the input is not a question (KEY is omitted), it is given to the function LERN which is responsible for committing it to memory if it is not already there explicitly. LERN first checks to see if RELATION appears in a paren pair on the paren list of NAME1. If it does not, a comma list named "/n" containing NAME2 is generated and the paren pair "(RELATION-/n)" is added to the paren list. If RELATION already appears, and the associated comma list does not contain NAME2, it is added to the comma list; otherwise, LERN does nothing. Thus the input NAME1 RELATION NAME2 is learned only if it does not already appear explicitly in memory. New paren pairs are put on the beginning of the paren list; if the paren list grows too long, the oldest pair on the list is deleted.

LERN is also given the reverse of the input statement allowing the program to take full advantage of the information content of the input. For example, the statement "MAX IS.SPOUSE.OF WENDY" causes the creation of a structure which can be represented graphically as follows: (See Appendix B)



If one views interrogation of such a structure as "pointer following", then entrance into the structure at the node "MAX" leads to "WENDY" via the relation "IS.SPOUSE.OF". On the other hand, entrance at the node "WENDY" leads to a dead end since the paren list for "WENDY" contains nothing to indicate that a pointer originating at "MAX" terminates at "WENDY". Supplying LERN with the statement "WENDY IS.SPOUSE.OF(INV) MAX"allows the above structure to be filled out by taking full advantage of the information content of the original statement. The final structure can be represented as:



It is important to note that the program is not being given any information concerning the relation of "IS.SPOUSE.OF" to the relation "IS.SPOUSE.OF(INV)". Thus, for instance, the program does not know that the relation "IS.SPOUSE.OF" is symmetric.

The first type of learning performed by the system is the simple committing to memory of a relation joining two nodes.

Example

before learning:

```
TOM::= (GENDER-/1)(IS.OFFSPRING.OF-/2)

JANE::= (GENDER-/4)(IS.SPOUSE.OF-/5)
```

after learning TOM LIKES JANE:

```
TOM::= (LIKES-/3)(GENDER-/1)(IS.OFFSPRING.OF-/2)

JANE::= (LIKES(INV)-/6)(GENDER-/4)(IS.SPOUSE.OF-/5)

  /3 ::= JANE,

  /6 ::= TOM,
```

This type of learning includes as a special case the learning of alternate definitions of a relation. These alternate definitions are used during execution of the question answering routines. For the present, "EQUIV" may be viewed as just another relation. Suppose part of the memory structure is

IS.GRANDFATHER.OF :: = (IS.MEM-/1)(EX.IS-/2)

after being given the input statement

"IS.GRANDFATHER.OF EQUIV (GENDER-MALE)

IS.PARENT.OF/IS.PARENT.OF"

the memory structure is

IS.GRANDFATHER.OF :: = (EQUIV-/3) (IS.MEM-/1) (EX.IS-/2)

/3 :: = (GENDER-MALE) IS.PARENT.OF/IS.PARENT.OF/,

3.2.2 IMPLICIT LEARNING.

The second type of learning occurs when, in the process of answering a question, the program generates pointers which are not explicitly in the net. Thus, after answering the question using implicit information, the program makes the answer explicit so that the next time a question is asked, it will be easier to answer.

Example

Structure before interrogation

```
        JOHN ::= (IS.FATHER.OF-/1) (GENDER-/2)
        /1    :: = FRANK
        FRANK ::= (IS.FATHER.OF-/3) (GENDER-/4)
        /3    :: = MIKE,
        IS.GRANDFATHER.OF :: = (EQUIV-/5) (IS.MEM-/6)
        /5    :: = IS.FATHER.OF/IS.FATHER.OF/,
```

structure after being asked JOHN IS.GRANDFATHER.OF WHO ***

```
        JOHN ::= (IS.GRANDFATHER.OF-/7) (IS.FATHER.OF-/1)(GENDER-/2)...
        /7    :: = MIKE,
```

In answering this question, the program made use of the fact that the relation IS.GRANDFATHER.OF is equivalent to the composition of the relation IS.FATHER.OF with itself. This information enables the question answering routine to find JOHN's grandfather even though this information did not appear explicitly anywhere in the net. To avoid the equivalence search upon future occasions, the program makes the grandfather pointer explicit on JOHN's paren list.

3.2.3 LEARNING BY EXAMPLE

The third type of memory modification involves the discovery of relational definitions implicitly contained in the net. As an example consider the truncated genealogy chart given below. The explicit relation connecting MAX and TOM is IS.PARENT.OF/IS.PARENT.OF/. Suppose the question "WHO

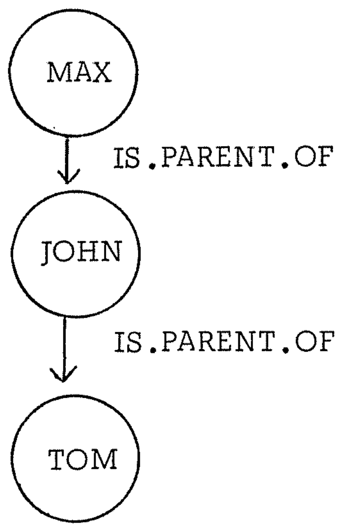
IS.GRANDFATHER.OF TOM" is asked. If we assume that the program is unfamiliar with the realtion IS.GRANDFATHER.OF then what action can the program take? One alternative, naturally, is to inform the interrogater that IS.GRANDFATHER.OF is not in the program's vocabulary; however, there may be sufficient information in the net to define IS.GRANDFATHER.OF in terms of relations known to the net. This is what the function CONJECTURE does. Given a small amount of information by the interrogater, CONJECTURE attempts to define the unknown relation in terms of known relations and node properties. If it succeeds, it then places the definition on the <u>equiv list</u> of the new relation in the format previously described. (If necessary, this list is created furing the execution of CONJECTURE.)

<u>Example of the operation of CONJECTURE</u>

For the relation IS.GRANDFATHER.OF, CONJECTURE generates the <u>equiv list</u>

IS.GRANDFATHER.OF :: = (EQUIV-/1)

/1 :: = (GENDER-MALE) IS.PARENT.OF/IS.PARENT.OF



In the process of discovering a definition of IS.GRANDFATHER.OF, the executive portion of the program gives intermediate results including MAX IS.GRANDFATHER.OF TOM to the LERN function. Thus, newly learned relations are inserted locally into the net and the new definition is globally available via the equiv list for future question answering. (see p. 21)

CONJECTURE's operation.

CONJECTURE allows the program to learn by example -or somewhat suggestively- generalize by extracting those characteristics that are common to all examples of an example set. Thus, presented with the problem "WHO IS.GRANDFATHER.OF TOM", CONJECTURE would respond by asking for the following information.

1. Some examples of known node names connected by the relation IS.GRANDFATHER.OF (consists of a list of name pairs)
2. The class to which IS.GRANDFATHER.OF belongs (FAMILY RELATION)
3. the allowable maximum number of known relations which may be composed to form a definition of IS.GRANDFATHER.OF.

(Note that although 2 & 3 are supplied explicitly in the present program, it is hoped that they will eventually become learned parameters.)

CONJECTURE consists of three major parts: DISCVR, COMPROP, and HYPGEN. DISCVR takes the two names comprising the first example and discovers all connecting paths within the net consisting of relations in the prescribed class (in this case FAMILY.RELATION contains IS.SPOUSE.OF, IS.OFFSPRING.OF, IS.SIBLING.OF and IS.PARENT.OF.) The length of any of these paths may not exceed the prescribed maximum. HYPGEN calls the routine CULL and tests the discovered paths' validity on the remaining examples. The residual paths form part of the new definition of the unknown relation. HYPGEN then supplies the examples to COMPROP which determines common node properties as follows: COMPROP accepts the first name of the first example part and makes a copy of its _paren list_. From this copy it then deletes all _paren pairs_ containing a relation component belonging to CLASS (since these have already been accounted for by DISCVR.) It then determines which of the relational components of the remaining _paren pairs_ appear on all the _paren lists_ associated with first names of the examples. For each such relation, all associated _comma lists_ are intersected yielding a series of _paren pairs_ having the property that each relational component appears on every first name _paren list_ of each example. The associated _comma list_ contains only those elements common to all the appropriate _comma lists_. The _paren list_, which results, forms the first node requirements that were mentioned previously. The last names in the example set are processed in a similar manner. (Future versions of the program will have the ability to generate properties for intermediate nodes.) As an example of the operation of COMPROP consider the following:

Input given to the program is:

JOHN (relation) MARY

TOM (relation) JANE

```
JOHN ::= (HAS-/1) (LIKES-/2)     where /1::= LEGS, HEAD, CAR, BIKE
TOM  ::= (HAS-/3) (GENDER-/4)    where /3::= LEGS, HEAD, HOUSE, STORE,
                                              TEMPER
```

The value returned by COMPROP relating to the first nodes is then:

(HAS - LEGS, HEAD,)

3.3 QUESTION ANSWERING

If the input to the program is found to be a question, it is given to the function ANSWR to determine what type of question it is and to delegate it to the appropriate question answering function. There are four question-types, falling into two categories:

I. Verify NAME1 RELATION NAME2 and answer true or false.

II. Fill in the blank. Answer with a list of nodes or relations.

A. NAME1 RELATION __________. NAME1 is related to which nodes by RELATION?

B. __________ RELATION NAME2. Which nodes are related to NAME2 by RELATION?

C. NAME1 __________ NAME2. How are NAME1 and NAME2 related?

Examples:

I. JOHN LIKES MARY ***

II.

A. JOHN LIKES WHO ***

B. WHO LIKES JOHN ***

C. DAVID HOW.RELATED WENDY ***

The "***" key indicates that a given input is a question. To determine which type of question it is, the function ANSWR checks the paren list for each element of the triple that makes the question, looking for the relation "IS". If it appears then its comma list is searched for "QUESTION". If "QUESTION" also appears, then the question element upon whose paren list we are looking is the interrogative element for the question, corresponding to the blank in the question-type definition above. Thus any name may function as a question word as long as the program has previously been told that it is a question word. In the present example, the program must have previously encountered the statement "WHO IS QUESTION".

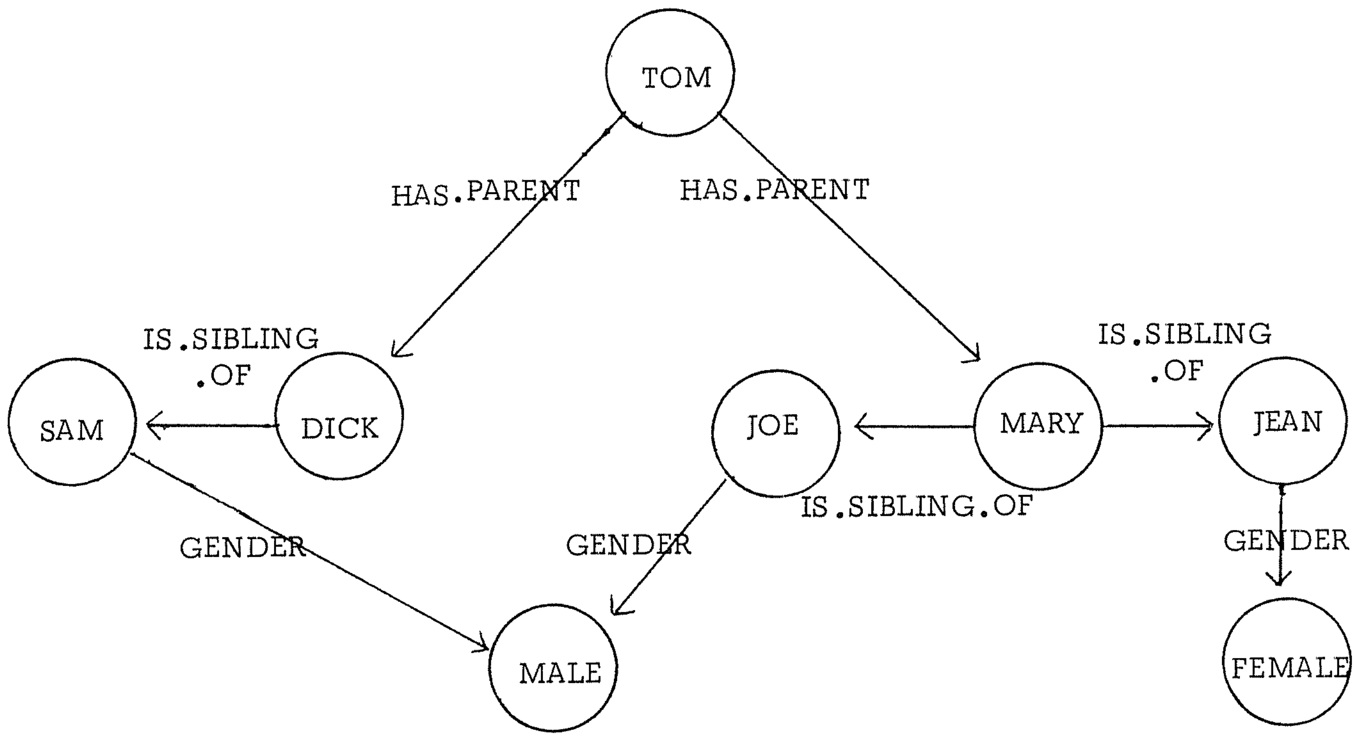
ANSWR now has sufficient information to know what type of question is being asked and thus delegates the question to the appropriate question answering function. The answers delivered by the search procedures can be culled by eliminating any node which does not satisfy the requirements of the interrogative word in the question. For instance, the answer to a WHO question must be a person.

SSTRFL and DCMPSTRFL are used to answer all of the question types except type II.C. which is answered by DESCEND. The following sections will present a detailed discussion of each of these functions.

3.3.1 SSTRFL

SSTRFL is given two parameters, a name and a S rel name. Its job is to give a comma list such that each comma list element is related to the parameter name by the relation represented by the parameter S rel name. For instance if memory contained:

```
TOM::= (HAS.PARENT-/1)
DICK::= (IS.SIBLING.OF-/2)
MARY::= (IS.SIBLING.OF-/3)
SAM::= (GENDER-/4)
JOE::= (GENDER-/4)
JEAN::= (GENDER-/5)
/1::= DICK, MARY,
/2::= SAM,
/3::= JOE, JEAN,
/4::= MALE,
/5::= FEMALE.
```


TOM
HAS.PARENT
HAS.PARENT
IS.SIBLING .OF
IS.SIBLING .OF
SAM
DICK
JOE
MARY
JEAN
IS.SIBLING.OF
GENDER
GENDER
GENDER
MALE
FEMALE

and SSTRFL were given the arguments TOM and HAS.PARENT/IS.SIBLING.OF/, it should return SAM, JOE, JEAN.

This explicit string following is accomplished by SSTRFL's giving to STRFOL a name and a <u>rel name</u> at a time and building up the answer.

STRFOL operated upon <u>name</u> with <u>rel name</u> and produces a <u>comma list</u> of nodes related to <u>name</u> by <u>rel name</u>. In this example SSTRFL would first give STRFOL, TOM and HAS.PARENT and get back DICK, MARY. Then it would give STRFOL DICK and IS.SIBLING.OF and get SAM, and finally give STRFOL MARY and IS.SIBLING.OF and get JOE, JEAN.

What if now, with the same memory as above, SSTRFL were given TOM and UNCLE? STRFOL would be given TOM and UNCLE and would fail since UNCLE is not a <u>rel name</u> on TOM's <u>paren list</u>. SSTRFL would now try to answer its question using implicit information. It would therefore call DCMSTRFL to make use of alternate definitions of UNCLE, but if memory had none at this time, DCMPSTRFL would fail, and SSTRFL would call CONJASK to determine the desire of the interrogator. Suppose CONJECTURE is called, causing the following lists to be generated and then entered into memory:

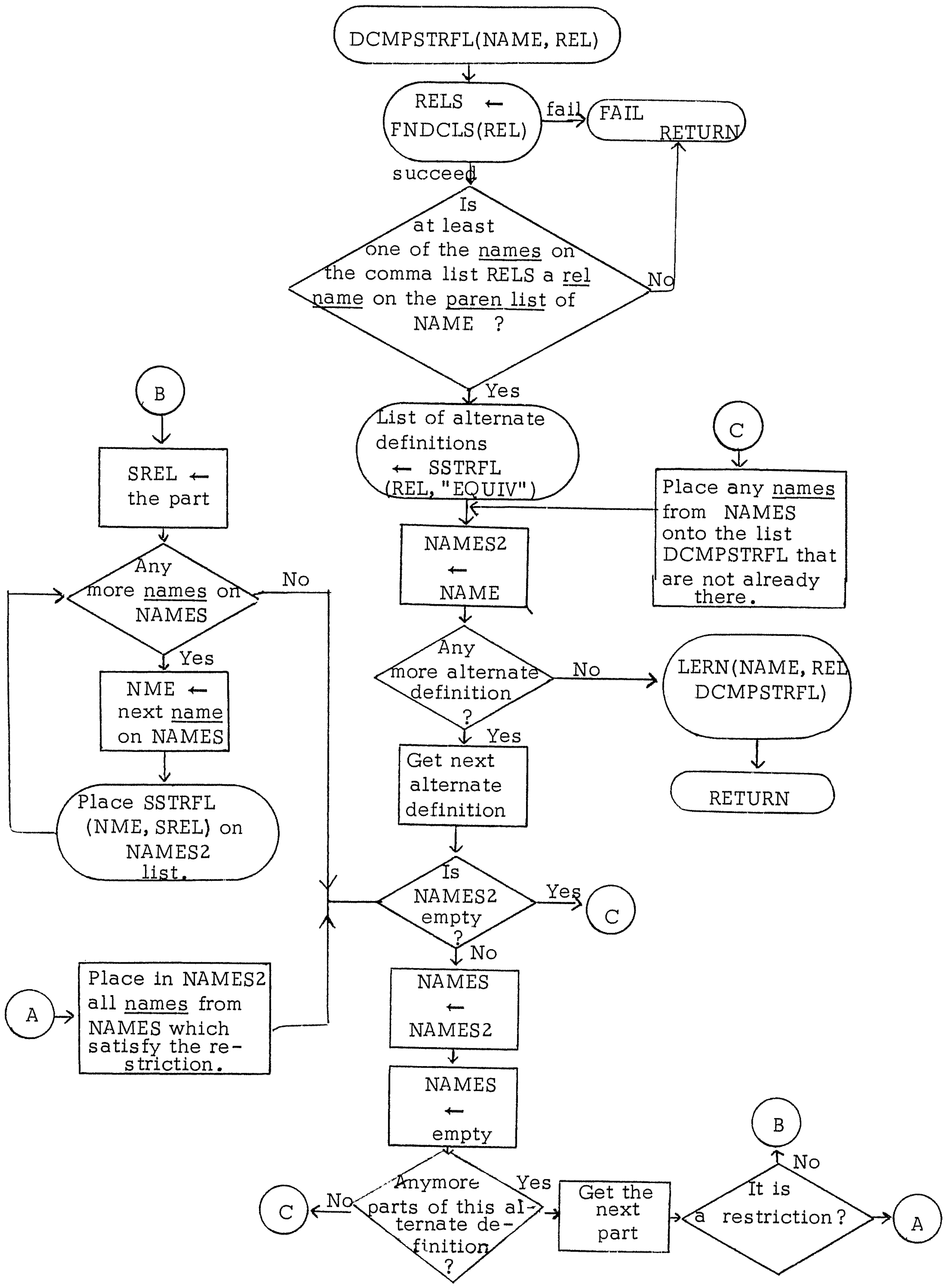
```
UNCLE::= (EQUIV-/6)
/6::= HAS.PARENT/IS.SIBLING.OF/(GENDER-MALE)
```

Now SSTRFL can again call DCMPSTRFL with the arguments TOM and UNCLE. This time DCMPSTRFL will succeed and with some recursive calls to SSTRFL will return the answer SAM, JOE which SSTRFL will return as its result.

3.3.2 DCMPSTRFL

DCMPSTRFL takes two arguments, a name and a rel name. Its job is like SSTRFL and STRFOL in that it is to return a comma list of names that satisfy the argument relation on the argument name. DCMPSTRFL knows, however, that the rel name is not on the name's paren list and it must find alternate definitions and use them. If the alternate definitions exist they will be found by giving SSTRFL the argument rel name and the rel name EQUIV. SSTRFL will return the equiv list on which each equiv list element is one definition. For example, if it were given UNCLE and EQUIV/ in the example above it would return HAS.PARENT/IS.SIBLING.OF/ (GENDER-MALE), . There may, of course, be several alternate definitions on the equiv list.

DCMOSTRFL applies all alternate definitions to the argument name so that as complete an answer as possible may be given. It applies each definition by taking each equiv list part in turn and applying it to all the names on a work list. To see how this is done we shall follow the example given above, namely giving TOM and HAS.UNCLE to DCMPSTRFL. DCMPSTRFL first makes sure that HAS.UNCLE has alternate definitions, and then checks to see if it might be possible to get an answer using them. The reasoning behind this second check is that we already know that the initial rel name (HAS.UNCLE in this case) is not on the name's (TOM's) list, so if we are to succeed, the first rel name on at least one of the definitions must be on the name's list (in this case HAS.PARENT is on TOM's paren list,


DCMPSTRFL(NAME, REL)
RELS ← FNDCLS(REL)
fail
FAIL RETURN
succeed
Is at least one of the names on the comma list RELS a rel name on the paren list of NAME ?
No
Yes
List of alternate definitions ← SSTRFL (REL, "EQUIV")
NAMES2 ← NAME
Any more alternate definition ?
No
LERN(NAME, REL DCMPSTRFL)
RETURN
Yes
Get next alternate definition
Is NAMES2 empty ?
Yes
C
No
NAMES ← NAMES2
NAMES ← empty
Anymore parts of this alternate definition ?
No
C
Yes
Get the next part
It is a restriction ?
No
B
A
C
Place any names from NAMES onto the list DCMPSTRFL that are not already there.
B
SREL ← the part
Any more names on NAMES
No
Yes
NME ← next name on NAMES
Place SSTRFL (NME, SREL) on NAMES2 list.
A
Place in NAMES2 all names from NAMES which satisfy the restriction.

so we may proceed). If none of them are, then at least one of them must have an alternate definition whose first rel name is on the paren list. If none of them are there, then we cannot succeed. We may even get into an infinitely recursive process of trying alternate definitions for example if all kinship relations were defined in terms of each other and we asked for the nephew of Gone With the Wind, or the grandfather of Adam.

To help in these tests, DCMPSTRFL immediately calls FNDCLS (which is helped by ENDCLS1) giving it the rel name as its argument. FNDCLS first gets hold of the rel name's equiv list by calling STRFOL with the rel name and EQUIV. If STRFOL fails because EQUIV is not on the rel name's paren list then FNDCLS fails causing DCMPSTRFL to fail indicating to SSTRFL that the rel name cannot be redefined. If the equiv list is found, FNDCLS forms a comma list of all rel names that would be applied first if any of the alternate definitions were used. In our example above this would just consist of HAS.PARENT, since HAS.PARENT has no alternate definition. If HAS.UNCLE were defined in terms of HAS.FATHER and HAS.MOTHER and each of them were defined in terms of HAS.PARENT, the list would be HAS.FATHER, HAS.MOTHER, HAS.PARENT. This process is finite even in the case of a circular definition since the alternate definitions of a rel name are looked at only once, and if a rel name is already on the list it is not added again. FNDCLS causes this list to be learned by calling LERN with the rel name, CLASS and the comma list as arguments. In our example the value of HAS.UNCLE would now be (CLASS-/7) (EQUIV-/6) and the value of /7 would be HAS.PARENT. DCMPSTRFL takes the comma list provided by FNDCLS and makes sure that at least one rel name on it is on the paren list of the argument name it was given. If not, it fails.

If all the tests succeed, DCMPSTRFL gets the equiv list and starts applying it to the argument name. It first places the argument name on a work list, and takes the first equiv list element (the first definition). It takes the first equiv list part of this element and if it is an S rel name gives the name on the work list and this rel name to SSTRFL. This, of course, might cause further recursive calls to DCMPSTRFL. In our example the first equiv list part is the S rel name HAS.PARENT/IS.SIBLING.OF/, so after SSTRFL is called the work list will be changed to SAM, JOE, JEAN. At this point the work list is given to CNDNS to make sure no name appears more than once. If there is now another equiv list part it will be an equiv paren pair. This acts as a restriction, DCMPSTRFL purging from the work list any name which does not fulfill its requirement. In our example this equiv paren pair is (GENDER-MALE,), so the only names which stay on the work list are those with GENDER on their paren list and MALE on the comma list associated with it. Thus after this operation is performed, the work list is SAM, JOE, . If there is another equiv list part it could be either an equiv paren pair or an S rel name, and the work list is appropriately adjusted. When all equiv list parts are used the work list contains an answer, but possibly only a partial answer. If there are other definitions on the equiv list they are also applied to the argument name, and all the final work lists are put on one comma list, which is given to CNDNS and returned as the answer. Before DCMPSTRFL returns, however it gives to LERN the argument name, the argument rel names and the answer comma list, so that if this information is required again, it will be readily available. In our example the final answer list is SAM, JOE, so TOM's value is changed to (HAS.UNCLE-/8) (HAS.PARENT-/2) and the slash name "/8" is added to memory with the value SAM, JOE, .

3.3.3 DESCEND

DESCEND is a general search procedure which could be used for any type of search. At present it is used solely for finding the paths joining two nodes. It has four parameters: the two nodes, a relation, and the maximum number of levels the search is to descend. The relation "ANY" is used as the relation parameter when we are just finding paths from A to B .

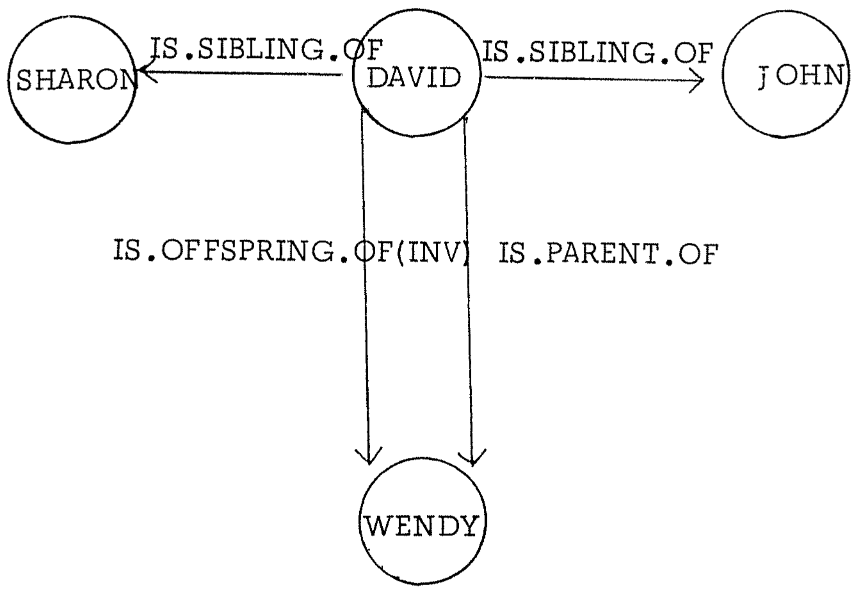
This relation parameter serves a purpose analogous to that of the CLASS parameter in CONJECTURE. In this case the value "ANY" corresponds to the universal class.

The search strategy is a level by level search from each of the nodes. At each level the nodes reached from the right are compared to those reached from the left. If there is a common node, we have found c omplete path from one node to the other. The parallel level by level search is faster than a search using just one of the nodes. If there are N paths leading from every node, then after finding paths joining two nodes of length M the onesided search has tried N^M paths. The parallel search, on the other hand, has tried only $2N^{M/2}$ different paths, a much smaller number.

DOWN

Down is the function called by DESCEND to extend the search one level more away from one node. Its only parameter is a string of paths leading away from a starting node. So if DAVID were the starting node, and

```
DAVID ::= (IS.OFFSPRING.OF(INV)-/1) (IS.SIBLING.OF-/2)
    (IS.PARENT.OF-/3)
/1     ::= WENDY,
/2     ::= JOHN, SHARON,
/3     ::= WENDY,
```



DOWN would then return four incomplete paths: the one leading from DAVID to JOHN, the one from DAVID to SHARON, and the two from DAVID to WENDY. Since DOWN extends the paths leading away from only one of the two nodes given to DESCEND as parameters, it must be called twice to extend the search one level, once for each side. DOWN proceeds by taking paren pairs off the paren list and decides to follow the relation after it is accepted by RELEVANTK.

RELEVANTK

RELEVANTK is potentially the most interesting part of the search. DESCEND is potentially a worst case search meaning that it is prepared to follow all paths indiscriminantly. RELEVANTK is charged with the responsibility of keeping the search within certain bounds. If DESCEND is given "ANY" as its relation parameter, then RELEVANTK will allow DOWN to follow all relations except those eliminated by PRUNE (described next). If the relation leading away from a node is the same given to the DESCEND as a parameter, then it is always accepted. Then if the question asking how two nodes are related is asked with respect to a given relation or class of relations, RELEVANTK asks itself whether the relation being considered is related to the relation given to DESCEND as a parameter. In order to do this, it assumes that relations are described by paren lists in the same way as other information. So the relation HAS.FATHER would have the paren list:

```
HAS.FATHER ::= (IS.MEM-/1) (EQUIV-/2)(INV-/3)(EX.IS-/4)
/1          :: = FAMILY.RELATIONS,
/2          :: = HAS.PARENT/(GENDER-MALE)
/3          :: = IS.OFFSPRING.OF, SON, DAUGHTER,
/4          :: = JOHN, MAX, DAVID,
```

So, if the question asked was how are two nodes connected with family relations, and the path being considered is "HAS.FATHER", then RELEVANTK calls DESCEND again with FAMILY.RELATIONS and HAS.FATHER as node parameters and "ANY" as the relation parameter to see if the HAS.FATHER path is relevant to a search for family relations. (Remember we may already be several layers down in DESCEND.) In the above example, DESCEND would find that there is a path from HAS.FATHER to FAMILY.RELATIONS consisting of only one step and so the HAS.FATHER path would be followed.

PRUNE

PRUNE checks the relation being considered to see if it appears on a list of dead end relations, ie. those paths which should not be followed. At present there is a global list called DEAD.END.LST which contains the names of all paths known to be not worth following. An example of such a path would be GENDER which will point to either male or female which will in turn point to all examples of males and females respectively. Obviously, there is little to be gained by including the set of all humans in a selective search involving family relations. In the future it may be possible for the program to learn which paths are never fruitful or if they are, yield trivial results. Hopefully, the program could learn to identify the dead end words associated with a particular context and apply them only to searches made within that context.

4.0 SUGGESTED PROGRAM EXTENSIONS

As mentioned in the introduction, our major interest in the semantic question answering problem lies with the memory structure and the functions necessary to interrogate, expand and reorganize that structure. In section 3.0 we viewed these tasks as naturally falling into the categories: Memory Structure, Learning and Question Answering.

At this point we feel that the semantic associational net structure adopted herein is an adequate basis for future extensions of the program. Thus, we do not foresee any significant changes in the Memory Structure as described in Appendices A & B. The learning portion of the program and the question answering executive on the other hand, are areas in which we anticipate further activity.

4.1 ADDITIONS TO THE QUESTION ANSWERING EXECUTIVE

A learning question answering program, by necessity, is an interactive program. In fact, the "learning by example" portion of the program presupposes that the program can interrogate its environment. Since this is not possible with the present SNOBOL configuration, such responses on the part of the program are anticipated by the programmer and presupplied by applicable example sets. To utilize this capability of the program, it would be necessary to extend the current SNOBOL implementation to allow Teletype communication with the program during a run. In the meantime, it is possible to simulate the relevant aspects of interaction as we have done. This allows us to examine those interactive modes of operation applicable to a question answering program.

Because the program is capable of utilizing alternate definitions of a relation, it is possible to respond to a query with varying amounts of detail. Thus, for instance, suppose the program was asked for Max's offspring. If Max had only one offspring explicitly represented on his paren list but had several implicitly represented (say, via his spouse), then the existing program would respond with only the explicitly represented offspring. Since offspring could conceivably have the alternate definition IS.SPOUSE.OF/ HAS.OFFSPRING/, the program has the necessary information to expand on the answer. Ideally, in an interactive mode, the interrogator could indicate that he desired more information. This could cause the program to dig deeper by forcing it to also utilize any alternate definitions it might have of the given relation.

4.2 ADDITIONS TO THE LEARNING PORTIONS OF THE PROGRAM

The equiv list is a very important feature of the current program. Since the program learns the EQUIV(INV) list at the same time it learns the equiv list of a relation, it would be useful to provide some mechanism by which the EQUIV(INV) list could be utilized. Currently this is not done. In the future version of the program we anticipate the inclusion of a facility to allow for the introduction of synonyms for EQUIV. Thus, the program might consider (once it had been taught) EQUIV(INV), EQUAL, SAME.AS as all being equivalent to EQUIV. The program, in trying to answer a question, would then use these relations in exactly the same manner as EQUIV is now used, thus bringing to bear the powerful recursive features of DCMPSTRFL, SSTRFL and STRFOL. Inclusion of this capability would effect a certain degree of generality in allowing the program to utilize existing information available via EQUIV(INV) as well as enhancing readability by allowing user supplied synonyms for EQUIV.

When asked a question, the program generally learns new facts in the process of answering the question. The one exception is in the case where the answer to the question is explicitly contained the net. In this case no new pointers are generated and learning does not take place. This is unfortunate since the information contained in a question indicates a portion of the net structure that is relevant to the external environment, thus implying that that portion of the structure is important and should be developed. We hope to implement this type of building in the future version of the program, in conjunction with the increased question answering capability mentioned above. This would cause any intermediate information to be assimilated into the net in the vicinity of the names appearing in the question. Additionally, the program might be forced to ask itself questions relevant to these "interesting" portions of the net. In this case intermediate information discovered in the process of answering such "introspective" questions would be incorporated into the net as before.

The COMPROP portion of CONJECTURE provides the capability for a limited type of generalization in the sense that an example set of the form X1 Y, X2 Y, ... Xn Y where Y was a dummy node and all pairs were related by the dummy relation ALPHA, would cause ALPHA to stand for the set of properties common to the names X1, X2, ..., Xn. It would be interesting to expand on this capability by allowing for disjunctive relations among the names X1, ..., Xn as well as allowing for cross relations between the common properties of left and right hand nodes. As an example, the relation KISSING.COUSINS EQUIV (GENDER-MALE) COUSIN/(GENDER-FEMALE), (GENDER-FEMALE)COUSIN/(GENDER-MALE), could not be learned implicitly by the present program from the example set JOE MARY, SAM ALICE, WENDY TOM, LOREE JOHN, since the gender information would be lost during the execution of COMPROP. To handle this example set, it would be necessary for the CONJECTURE function to group the first two examples to obtain (GENDER-MALE) COUSIN/(GENDER-FEMALE) and group the last two to obtain (GENDER-FEMALE) COUSIN/(GENDER-MALE).

In the current version of the program, the depth and class parameters for CONJECTURE are supplied explicitly in the examples. Eventually these should be learned. FNDCLS could be helpful in this respect, since the classes it forms are similar to those used as values for the CLASS parameter.

The current program contains two functions PSIZE and DELPAR which are responsible for keeping <u>paren lists</u> within manageable bounds. This is accomplished by deleting a <u>paren pair</u> from the end of a <u>paren list</u> that has grown too large. We should, however, try to insure that the more important <u>paren pairs</u> are not deleted. This was a motivation for the function REINF which moves a <u>paren pair</u> closer to the front of a <u>paren list</u>. Unfortunately, we have not yet formed a satisfactory method of determining when and where to apply it.

Our program, as the reader has certainly realized, is based on binary relations. That is, relations between two objects. Although this takes care of a large interesting class of relations, there are others that we would like to handle. For example, sentences like "John gave the book to Mary" and "John, Joan, Jim, and Bill are friends."
Although these may be reduced to a conjunction of binary relations, it is not clear how they may be conveniently handled by the existing memory structure without some form of preprocessing.

Finally, a learning system always runs the risk of learning something that is incorrect. Our program is particularly vulnerable to misinformation during the execution of CONJECTURE. Example sets that are atypical or those that provide misleading information because they are too small, introduce the possibility of error propagation throughout the net. Because no satisfactory semi-automatic purging procedures are currently operable, we have partially protected the memory structure by having it assimilate only the local part of new information obtained by CONJECTURE while keeping the global part separate on the appropriate <u>equiv list</u>. Thus if the relation

"IS.GRANDFATHER.OF" were not in memory, but the statements "MAX IS.PARENT.OF JOHN" and "JOHN IS.PARENT.OF TOM" had previously been learned and the question "WHO IS.GRANDFATHER.OF TOM" were asked, the memory structure would be modified as follows:

1) A new paren pair containing the relation "IS.GRANDFATHER.OF" would be added to MAX's paren list and would point via the associated comma list to "TOM".

2) A new list named "IS.GRANDFATHER.OF" would be created containing a paren pair composed of the rel name EQUIV and a slash name. The comma list corresponding to the slash name would have an alternate definition of "IS.GRANDFATHER.OF" in terms of IS.PARENT.OF in the format previously described.

The "IS.GRANDFATHER.OF pointer from MAX to TOM is regarded as local information whereas the newly obtained alternate definition of "IS.GRANDFATHER.OF" is regarded as global information. At this point it would be possible to scan memory for all occurences of nodes connected by IS.PARENT.OF/IS.PARENT.OF/ superrelations and connect them with the simple relation IS.GRANDFATHER.OF . This is not done however, since it would be quite messy to reverse the process if the alternate definition of IS.GRANDFATHER.OF was later found to be in error. Perhaps the most reasonable way to proceed would be to use newly generated alternate definitions on a probationary basis. Probationary alternate definitions would be taken into the fold once they had demonstrated their utility to the user.

Problems closely related to the questions of how and when to reward and punish in order to achieve desirable program characteristics still remain unsolved. Until satisfactory solutions are found, *paren pair* ordering, *paren list* length maintenance, CLASS parameter learning, and *equiv list* purging must remain arbitrary.

APPENDIX A

Description of Memory Structure.

Restricted characters are /,), (, -

Characters are all letters, numbers and special characters except restricted characters.

Numbers are decimal integers.

BNF

```
<name> ::= <character> | <character> <name>

<rel name> ::= <name> | <name> (INV)

<slash name> ::= / <number>

<paren pair> ::= (<rel name> - <slash name>)

<paren list> ::= <paren pair> | <paren pair> <paren list>

<comma list element> ::= <name> | <slash name>

<comma list> ::= <comma list element> , | <comma list element> ,
        <comma list>

<S rel name> ::= <rel name> / | <rel name> / <S rel name>

<equiv paren pair> ::= < paren pair> | (<rel name> - <comma list>)

<equiv list part> ::= <S rel name> | <equiv paren pair>

<equiv list element> ::= <equiv list part> | <equiv list part>
        <equiv list element>

< equiv list> ::= < equiv list element>, | <equiv list element> ,
        <equiv list>
```

Semantics of Named Structures

A name is used as the name of a paren list.

A slash name is used as the name of a comma list except as noted below.

A rel name represents a relation on and into the set of names and rel names and is used as the name of a paren list on which may be the paren pair (EQUIV- <slash name>) with the slash name being the name of an equiv list.

An S rel name represents a compound relation formed by function composition on the relations represented by the rel names forming the S rel name.

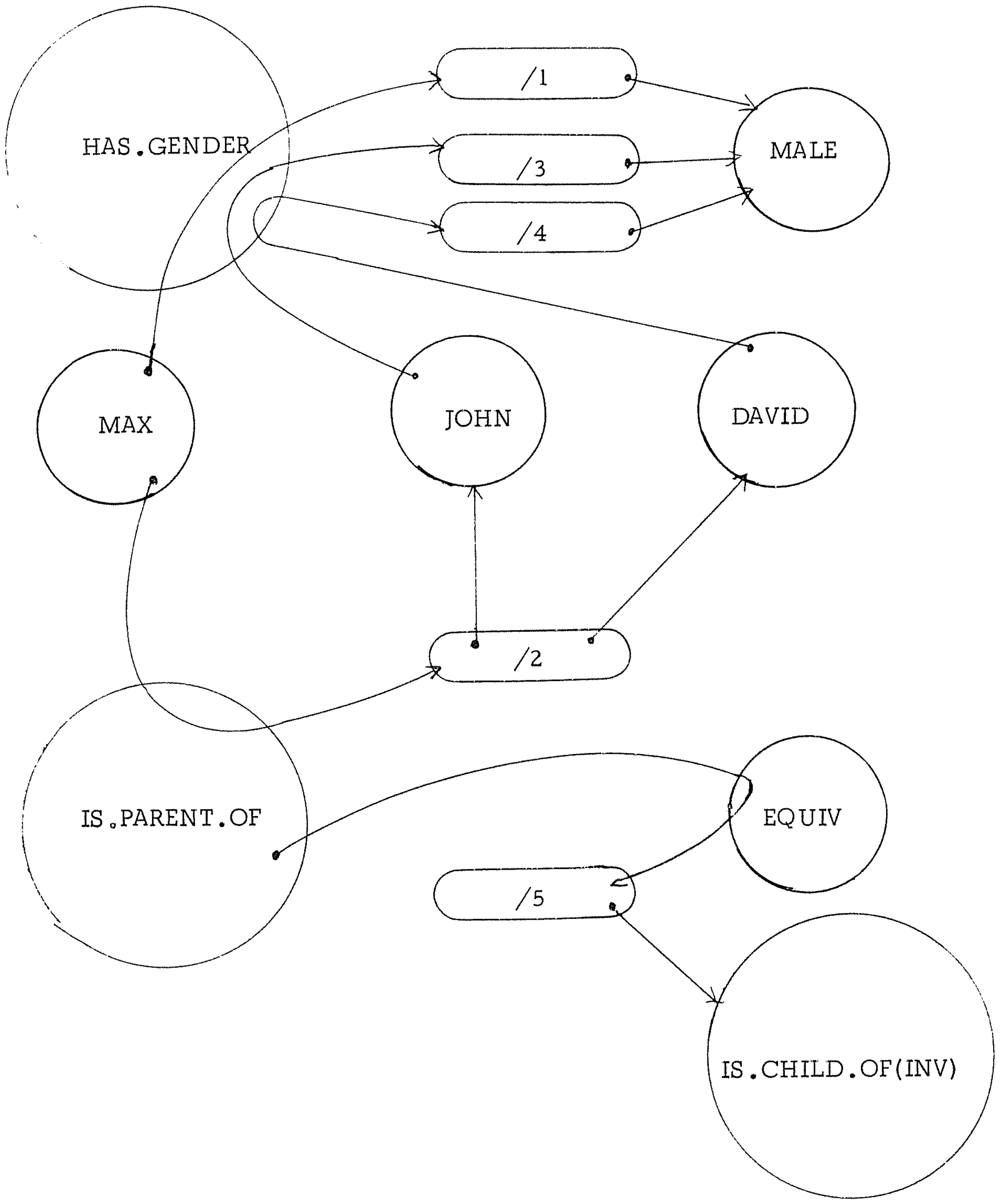
A comma list or equiv list is said to be associated with a rel name on a paren list if the rel name and the slash name whose value is the comma list are in the same paren pair on the paren list.

APPENDIX B

GRAPHICAL REPRESENTATION OF THE MEMORY STRUCTURE

There are three successively more complicated data structures used for complex information processing. These are lists, trees, and nets. Our data structure can, actually, be graphically represented best by something slightly more complicated than a net with labeled edges, since the labels (<u>rel names</u>) are also nodes (<u>names</u>). Therefore, an edge goes from a name through a <u>rel name</u> to a <u>slash name</u> which has pointers to the elements of its <u>comma list</u>. Thus, the figure below would represent the following section of memory:

```
MAX = (HAS.GENDER-/1)(IS.PARENT.OF-/2)

JOHN = (HAS.GENDER-/3)

DAVID = (HAS.GENDER-/4)

IS.PARENT.OF = (EQUIV-/5)

/1 = MALE,

/2 = JOHN, DAVID,

/3 = MALE,

/4 = MALE,

/5 = IS.CHILD.OF(INV),
```


HAS.GENDER
/1
/3
/4
MALE
MAX
JOHN
DAVID
/2
IS.PARENT.OF
EQUIV
/5
IS.CHILD.OF(INV)

APPENDIX C

Bibliography

1. Elliott, Roger W. A Model for a Fact Retrieval System, Ph.D. dissertation, University of Texas, 1965.

2. Feigenbaum, E. A. and Feldman, J. Computers and Thought, New York: McGraw-Hill, 1963.

3. Green, B. F., Wolf, A. K., Chomsky, C., and Laughery, K. "Baseball: An Automatic Question Answerer" in [2].

4. Lindsay, R. K. "Inferential Memory as the Basis of Machines which Understand Natural Language", in [2].

5. Levien, R. and Maron, M. E. Relational Data File: A Tool for Mechanized Inference Execution and Data Retrieval, Santa Monica, California: Rand Memorandum #RM-4793-PR, 1965.

6. Quillian, M. R. Semantic Memory, Ph.D. dissertation, Carnegie Institute of Technology, 1966. Also Bolt, Beranek and Newman Inc., Cambridge, Mass., 1966 for Air Force Cambridge Research Laboratories Report #AFCRL-66-189.

7. Raphael, B. SIR: A Computer Program for Semantic Information Retrieval, Ph.D. dissertation (Mathematics), M.I.T., 1964. Also Project MAC Report #TR-2.

APPENDIX D

A Sample Session Showing Some Uses of SAMENLAQ in a Literature Search Environment.

```
STATEMENT - WHO IS QUESTION
STATEMENT - WHAT IS QUESTION
STATEMENT - WROTE.WHAT.IN IS QUESTION
STATEMENT - AUTHOR.OF MEMBER BOOK.RELATIONS
STATEMENT - AUTHOR.OF(INV) MEMBER BOOK.RELATIONS
STATEMENT - PUBLISHED MEMBER BOOK.RELATIONS
STATEMENT - PUBLISHED(INV) MEMBER BOOK.RELATIONS
STATEMENT - DATED MEMBER BOOK.RELATIONS
STATEMENT - DATED(INV) MEMBER BOOK.RELATIONS
STATEMENT - REFERENCES MEMBER BOOK.RELATIONS
STATEMENT - REFERENCES(INV) MEMBER BOOK.RELATIONS
STATEMENT - APPEARED.IN MEMBER BOOK.RELATIONS
STATEMENT - APPEARED.IN(INV) MEMBER BOOK.RELATIONS
STATEMENT - E.HUNT AUTHOR.OF EXPERIMENTS.IN.INDUCTION
STATEMENT - J.MARIN AUTHOR.OF EXPERIMENTS.IN.INDUCTION
STATEMENT - P.STONE AUTHOR.OF EXPERIMENTS.IN.INDUCTION
STATEMENT - EXPERIMENTS.IN.INDUCTION DATED 1966
STATEMENT - ACADEMIC.PRESS PUBLISHED
            EXPERIMENTS.IN.INDUCTION
STATEMENT - EXPERIMENTS.IN.INDUCTION REFERENCES
            THE.SIMULATION.OF.VERBAL.LEARNING.BEHAVIOR
STATEMENT - E.FEIGENBAUM AUTHOR.OF
            THE.SIMULATION.OF.VERBAL.LEARNING.BEHAVIOR
STATEMENT - THE.SIMULATION.OF.VERBAL.LEARNING.BEHAVIOR APPEARED.IN
            PROC.WJCC
STATEMENT - THE.SIMULATION.OF.VERBAL.LEARNING.BEHAVIOR DATED
            1961
STATEMENT - EXPERIMENTS.IN.INDUCTION REFERENCES
            PERFORMANCE.OF.READING.TASK.BY.AN.EPAM.PROG
STATEMENT - E.FEIGENBAUM AUTHOR.OF
            PERFORMANCE.OF.READING.TASK.BY.AN.EPAM.PROG
STATEMENT - H.SIMON AUTHOR.OF
            PERFORMANCE.OF.READING.TASK.BY.AN.EPAM.PROG
STATEMENT - PERFORMANCE.OF.READING.TASK.BY.AN.EPAM.PROG APPEARED.IN
            BEHAVIORAL.SCI
STATEMENT - EXPERIMENTS.IN.INDUCTION REFERENCES
            CONCEPT.LEARNING
STATEMENT - E.HUNT AUTHOR.OF CONCEPT.LEARNING
STATEMENT - CONCEPT.LEARNING DATED 1962
STATEMENT - WILEY PUBLISHED CONCEPT.LEARNING
STATEMENT - EXPERIMENTS.IN.INDUCTION REFERENCES
            THE.MAGICAL.NUMBER.7+OR-2
STATEMENT - G.MILLER AUTHOR.OF THE.MAGICAL.NUMBER.7+OR-2
STATEMENT - THE.MAGICAL.NUMBER.7+OR-2 DATED 1956
STATEMENT - THE.MAGICAL.NUMBER.7+OR-2 APPEARED.IN PSYCH.REV.
STATEMENT - EXPERIMENTS.IN.INDUCTION REFERENCES
            STEPS.TOWARD.ARTIFICIAL.INTELLIGENCE
STATEMENT - M.MINSKY AUTHOR.OF
            STEPS.TOWARD.ARTIFICIAL.INTELLIGENCE
STATEMENT - STEPS.TOWARD.ARTIFICIAL.INTELLIGENCE DATED 1960
STATEMENT - STEPS.TOWARD.ARTIFICIAL.INTELLIGENCE APPEARED.IN
```

```
              PROC.IRE
STATEMENT  -  EXPERIMENTS.IN.INDUCTION REFERENCES IPLV.MANUAL
STATEMENT  -  A.NEWELL AUTHOR.OF IPLV.MANUAL
STATEMENT  -  IPLV.MANUAL DATED 1964
STATEMENT  -  PRENTICE.HALL PUBLISHED IPLV.MANUAL
STATEMENT  -  EXPERIMENTS.IN.INDUCTION REFERENCES GPS
STATEMENT  -  A.NEWELL AUTHOR.OF GPS
STATEMENT  -  H.SIMON AUTHOR.OF GPS
STATEMENT  -  GPS DATED 1961
STATEMENT  -  GPS APPEARED.IN LERNENDE.AUTOMATEN
STATEMENT  -  H.BILLING AUTHOR.OF LERNENDE.AUTOMATEN
STATEMENT  -  OLDENBOURG PUBLISHED LERNENDE.AUTOMATEN
STATEMENT  -  EXPERMENTS.IN.INDUCTION REFERENCES
              FIGHTS.GAMES.AND.DEBATES
STATEMENT  -  A.RAPOPORT AUTHOR.OF FIGHTS.GAMES.AND.DEBATES
STATEMENT  -  FIGHTS.GAMES.AND.DEBATES DATED 1960
STATEMENT  -  U.MICHIGAN PUBLISHED FIGHTS.GAMES.AND.DEBATES
STATEMENT  -  EXPERIMENTS.IN.INDUCTION REFERENCES
              COGNITION.AND.THOUGHT
STATEMENT  -  EXPERIMENTS.IN.INDUCTION REFERENCES PANDEMONIUM
STATEMENT  -  EXPERIMENTS.IN.INDUCTION REFERENCES
              COMPUTING.MACHINERY.AND.INTELLIGENCE
STATEMENT  -  A.TURING AUTHOR.OF
              COMPUTING.MACHINES.AND.INTELLIGENCE
STATEMENT  -  COMPUTING.MACHINERY.AND.INTELLIGENCE DATED 1950
STATEMENT  -  COMPUTING.MACHINERY.AND.INTELLIGENCE APPEARED.IN
              MIND
STATEMENT  -  J.KATZ AUTHOR.OF THE.STRUCTURE.OF.LANGUAGE
STATEMENT  -  J.FODOR AUTHOR.OF THE.STRUCTURE.OF.LANGUAGE
STATEMENT  -  PRENTICE.HALL PUBLISHED
              THE.STRUCTURE.OF.LANGUAGE
STATEMENT  -  THE.STRUCTURE.OF.LANGUAGE DATED 1964
STATEMENT  -  W.QUINE AUTHOR.OF
              THE.PROBLEM.OF.MEANING.IN.LINGUISTICS
STATEMENT  -  THE.PROBLEM.OF.MEANING.IN.LINGUISTICS APPEARED.IN
              THE.STRUCTURE.OF.LANGUAGE
STATEMENT  -  Z.HARRIS AUTHOR.OF DISTRIBUTIONAL.STRUCTURE
STATEMENT  -  DISTRIBUTIONAL.STRUCTURE APPEARED.IN
              THE.STRUCTURE.OF.LANGUAGE
STATEMENT  -  N.CHOMSKY AUTHOR.OF
              CURRENT.ISSUES.IN.LINGUISTIC.THEORY
STATEMENT  -  CURRENT.ISSUES.IN.LINGUISTIC.THEORY APPEARED.IN
              THE.STRUCTURE.OF.LANGUAGE
STATEMENT  -  N.CHOMSKY AUTHOR.OF
              ON.THE.NOTION.RULE.OF.GRAMMAR
STATEMENT  -  ON.THE.NOTION.RULE.OF.GRAMMAR APPEARED.IN
              THE.STRUCTURE.OF.LANGUAGE
STATEMENT  -  P.POSTAL AUTHOR.OF
              LIMITATIONS.OF.PHRASE.STRUCTURE.GRAMMARS
STATEMENT  -  LIMITATIONS.OF.PHRASE.STRUCTURE.GRAMMARS APPEARED.IN
              THE.STRUCTURE.OF.LANGUAGE
STATEMENT  -  N.CHOMSKY AUTHOR.OF
              A.TRANSFORMATIONAL.APPROACH.TO.SYNTAX
STATEMENT  -  A.TRANSFORMATIONAL.APPROACH.TO.SYNTAX APPEARED.IN
              THE.STRUCTURE.OF.LANGUAGE
STATEMENT  -  E.KLIMA AUTHOR.OF NEGATION.IN.ENGLISH
STATEMENT - NEGATION.IN.ENGLISH APPEARED.IN
            THE.STRUCTURE.OF.LANGUAGE
STATEMENT - M.HALLE AUTHOR.OF ON.THE.BASES.OF.PHONOLOGY
STATEMENT - ON.THE.BASES.OF.PHONOLOGY APPEARED.IN
            THE.STRUCTURE.OF.LANGUAGE
STATEMENT - M.HALLE AUTHOR.OF
            PHONOLOGY.IN.GENERATIVE.GRAMMAR
STATEMENT - PHONOLOGY.IN.GENERATIVE.GRAMMAR APPEARED.IN
            THE.STRUCTURE.OF.LANGUAGE
STATEMENT - Z.HARRIS AUTHOR.OF DISCOURSE.ANALYSIS
STATEMENT - DISCOURSE.ANALYSIS APPEARED.IN
            THE.STRUCTURE.OF.LANGUAGE
STATEMENT - N.CHOMSKY AUTHOR.OF DEGREES.OF.GRAMMATICALNESS
STATEMENT - DEGREES.OF.GRAMMATICALNESS APPEARED.IN
            THE.STRUCTURE.OF.LANGUAGE
STATEMENT - P.ZIFF AUTHOR.OF
            ON.UNDERSTANDING.UNDERSTANDING.UTTERANCES
STATEMENT - ON.UNDERSTANDING.UNDERSTANDING.UTTERANCES APPEARED.IN
            THE.STRUCTURE.OF.LANGUAGE
STATEMENT - J.KATZ AUTHOR.OF SEMISENTENCES
STATEMENT - SEMISENTENCES APPEARED.IN
            THE.STRUCTURE.OF.LANGUAGE
STATEMENT - R.CARNAP AUTHOR.OF
            FOUNDATIONS.OF.LOGIC.AND.MATHEMATICS
STATEMENT - FOUNDATIONS.OF.LOGIC.AND.MATHEMATICS APPEARED.IN
            THE.STRUCTURE.OF.LANGUAGE
STATEMENT - W.QUINE AUTHOR.OF SPEAKING.OF.OBJECTS
STATEMENT - SPEAKING.OF.OBJECTS APPEARED.IN
            THE.STRUCTURE.OF.LANGUAGE
STATEMENT - J.KATZ AUTHOR.OF
            THE.STRUCTURE.OF.A.SEMANTIC.THEORY
STATEMENT - J.FODOR AUTHOR.OF
            THE.STRUCTURE.OF.A.SEMANTIC.THEORY
STATEMENT - THE.STRUCTURE.OF.A.SEMANTIC.THEORY APPEARED.IN
            THE.STRUCTURE.OF.LANGUAGE
STATEMENT - W.REITMAN AUTHOR.OF COGNITION.AND.THOUGHT
STATEMENT - WILEY PUBLISHED COGNITION.AND.THOUGHT
STATEMENT - COGNITION.AND.THOUGHT DATED 1965
STATEMENT - COGNITION.AND.THOUGHT REFERENCES
            INTR.TO.THE.FORMAL.ANAL.OF.NATURAL.LANGUAGES
STATEMENT - INTR.TO.THE.FORMAL.ANAL.OF.NATURAL.LANGUAGES APPEARED.IN
            HANDBOOK.OF.MATH.PSYCH
STATEMENT - WILEY PUBLISHED HANDBOOK.OF.MATH.PSYCH
STATEMENT - HANDBOOK.OF.MATH.PSYCH DATED 1963
STATEMENT - COGNITION.AND.THOUGHT REFERENCES
            COMPUTERS.AND.THOUGHT
STATEMENT - E.FEIGENBAUM AUTHOR.OF COMPUTERS.AND.THOUGHT
STATEMENT - J.FELDMAN AUTHOR.OF COMPUTERS.AND.THOUGHT
STATEMENT - MCGRAW.HILL PUBLISHED COMPUTERS.AND.THOUGHT
STATEMENT - COMPUTERS.AND.THOUGHT DATED 1963
STATEMENT - COGNITION.AND.THOUGHT REFERENCES
            THE.SIMULATION.OF.VERBAL.LEARNING.BEHAVIOR
STATEMENT - E.FEIGENBAUM AUTHOR.OF
            THE.SIMULATION.OF.VERBAL.LEARNING.BEHAVIOR
STATEMENT - THE.SIMULATION.OF.VERBAL.LEARNING.BEHAVIOR
            APPEARED.IN
```

```
STATEMENT - THE.SIMULATION.OF.VERBAL.LEARNING.BEHAVIOR DATED
            1961
STATEMENT - COGNITION.AND.THOUGHT REFERENCES
            CONCEPT.LEARNING
STATEMENT - E.HUNT AUTHOR.OF CONCEPT.LEARNING
STATEMENT - WILEY PUBLISHED CONCEPT.LEARNING
STATEMENT - CONCEPT.LEARNING DATED 1962
STATEMENT - COGNITION.AND.THOUGHT REFERENCES
            EXPERIMENTS.IN.INDUCTION
STATEMENT - COGNITION.AND.THOUGHT REFERENCES IPLV.MANUAL
STATEMENT - COGNITION.AND.THOUGHT REFERENCES GPS
STATEMENT - COGNITION.AND.THOUGHT REFERENCES
            FIGHTS.GAMES.AND.DEBATES
STATEMENT - O.SELFRIDGE AUTHOR.OF PANDEMONIUM
STATEMENT - COGNITION.AND.THOUGHT REFERENCES PANDEMONIUM
STATEMENT - PANDEMONIUM DATED 1959
STATEMENT - COGNITION.AND.THOUGHT REFERENCES
            COMPUTING.MACHINERY.AND.INTELLIGENCE
STATEMENT - COGNITION.AND.THOUGHT REFERENCES
            PATTERN.RECOGNITION
STATEMENT - L.UHR AUTHOR.OF PATTERN.RECOGNITION
STATEMENT - PATTERN.RECOGNITION APPEARED.IN PSYCHOL.BULL
STATEMENT - PATTERN.RECOGNITION DATED 1963
STATEMENT - PSYCHOLINGUISTICS DATED 1961
STATEMENT - S.SAPORTA AUTHOR.OF PSYCHOLINGUISTICS
STATEMENT - HOLT.RINEHART.AND.WINSTON PUBLISHED
            PSYCHOLINGUISTICS
STATEMENT - LINGUISTICS.SYMBOLS.MAKE.MAN APPEARED.IN
            PSYCHOLINGUISTICS
STATEMENT - J.LOTZ AUTHOR.OF LINGUISTICS.SYMBOLS.MAKE.MAN
STATEMENT - ON.LINGUISTIC.TERMS APPEARED.IN
            PSYCHOLINGUISTICS
STATEMENT - F.HOUSEHOLDER AUTHOR.OF ON.LINGUISTIC.TERMS
STATEMENT - THE.INDEPENDENCE.OF.GRAMMAR APPEARED.IN
            PSYCHOLINGUISTICS
STATEMENT - N.CHOMSKY AUTHOR.OF THE.INDEPENDENCE.OF.GRAMMAR
STATEMENT - MEANING APPEARED.IN PSYCHOLINGUISTICS
STATEMENT - L.BLOOMFIELD AUTHOR.OF MEANING
STATEMENT - LANGUAGE.DEVELOPMENT.IN.CHILDREN APPEARED.IN
            PSYCHOLINGUISTICS
STATEMENT - J.CARROLL AUTHOR.OF
            LANGUAGE.DEVELOPMENT.IN.CHILDREN
STATEMENT - PHONEMIC.PATTERNING APPEARED.IN
            PSYCHOLINGUISTICS
STATEMENT - R.JAKOBSON AUTHOR.OF PHONEMIC.PATTERNING
STATEMENT - M.HALLE AUTHOR.OF PHONEMIC.PATTERNING
STATEMENT - APHASIA.AS.A.LINGUISTIC.PROBLEM APPEARED.IN
            PSYCHOLINGUISTICS
STATEMENT - R.JAKOBSON AUTHOR.OF
            APHASIA.AS.A.LINGUISTIC.PROBLEM
STATEMENT - CHOMSKY AUTHOR.OF
            INTR.TO.THE.FORMAL.ANAL.OF.NATURAL.LANGUAGES


 QUESTION - WHO AUTHOR.OF COGNITION.AND.THOUGHT
ANSWER - W.REITMAN
```

```
 QUESTION - N.CHOMSKY AUTHOR.OF WHAT
ANSWER - CURRENT.ISSUES.IN.LINGUISTIC.THEORY
   AND    ON.THE.NOTION.RULE.OF.GRAMMAR
   AND    A.TRANSFORMATIONAL.APPROACH.TO.SYNTAX
   AND    DEGREES.OF.GRAMMATICALNESS
   AND    THE.INDEPENDENCE.OF.GRAMMAR

 QUESTION - WHAT REFERENCES/AUTHOR.OF(INV) A.NEWELL
ANSWER - EXPERIMENTS.IN.INDUCTION
   AND    COGNITION.AND.THOUGHT

 QUESTION - EXPERIMENTS.IN.INDUCTION PUBLISHED.BY WHO

 WITH THE INFORMATION I NOW HAVE I CANNOT REDEFINE PUBLISHED.BY
SO THAT IT CAN BE APPLIED TO EXPERIMENTS.IN.INDUCTION.
IF YOU WISH TO ACCEPT THIS AS YOUR ANSWER ENTER - OK
IF YOU WISH TO SUPPLY AN ANSWER, ENTER A STATEMENT OF THE FORM -
 EXPERIMENTS.IN.INDUCTION PUBLISHED.BY X     OR      PUBLISHED.BY EQUIV X
OTHERWISE ENTER A SERIES OF PAIRS X Y SUCH THAT X PUBLISHED.BY Y
FOLLOWED BY THE CLASS OF RELATIONS PUBLISHED.BY BELONGS TO
FOLLOWED BY AN INTEGER INDICATING THE MAXIMUM NUMBER OF RELATIONS
THAT MAY BE COMPOSED TO GIVE AN ALTERNATE DEFINITION OF PUBLISHED.BY.
I WILL THEN ATTEMPT TO FIND A MORE SATISFACTORY  ANSWER TO YOUR QUESTION.

 STATEMENT - PUBLISHED.BY EQUIV PUBLISHED(INV)/
THANK YOU.

 I CAN NOT APPLY PUBLISHED.BY TO EXPERIMENTS.IN.INDUCTION
BUT I KNOW PUBLISHED.BY IS THE SAME AS
    PUBLISHED(INV)/
 I FIGURE EXPERIMENTS.IN.INDUCTION PUBLISHED.BY ACADEMIC.PRES
S,

ANSWER - ACADEMIC.PRESS

 QUESTION - WHAT DATED 1963
ANSWER - HANDBOOK.OF.MATH.PSYCH
   AND    COMPUTERS.AND.THOUGHT
   AND    PATTERN.RECOGNITION

 QUESTION - W.REITMAN REFERENCES WHAT

 WITH THE INFORMATION I NOW HAVE I CANNOT REDEFINE REFERENCES
SO THAT IT CAN BE APPLIED TO W.REITMAN.
IF YOU WISH TO ACCEPT THIS AS YOUR ANSWER ENTER - OK
IF YOU WISH TO SUPPLY AN ANSWER, ENTER A STATEMENT OF THE FORM -
 W.REITMAN REFERENCES X     OR      REFERENCES EQUIV X
OTHERWISE ENTER A SERIES OF PAIRS X Y SUCH THAT X REFERENCES Y
FOLLOWED BY THE CLASS OF RELATIONS REFERENCES BELONGS TO
FOLLOWED BY AN INTEGER INDICATING THE MAXIMUM NUMBER OF RELATIONS
THAT MAY BE COMPOSED TO GIVE AN ALTERNATE DEFINITION OF REFERENCES.
I WILL THEN ATTEMPT TO FIND A MORE SATISFACTORY  ANSWER TO YOUR QUESTION.

 STATEMENT - REFERENCES EQUIV AUTHOR.OF/REFERENCES/
THANK YOU.
```

```
 I CAN NOT APPLY REFERENCES TO W.REITMAN
BUT I KNOW REFERENCES IS THE SAME AS
     AUTHOR.OF/REFERENCES/

 I FIGURE W.REITMAN REFERENCES INTR.TO.THE.FORMAL.ANAL.OF.NAT
URAL.LANGUAGES,COMPUTERS.AND.THOUGHT,THE.SIMULATION.OF.VERBA
L.LEARNING.BEHAVIOR,CONCEPT.LEARNING,EXPERIMENTS.IN.INDUCTIO
N,IPLV.MANUAL,GPS,FIGHTS.GAMES.AND.DEBATES,PANDEMONIUM,COMPU
TING.MACHINERY.AND.INTELLIGENCE,PATTERN.RECOGNITION,

ANSWER - INTR.TO.THE.FORMAL.ANAL.OF.NATURAL.LANGUAGES
   AND    COMPUTERS.AND.THOUGHT
   AND    THE.SIMULATION.OF.VERBAL.LEARNING.BEHAVIOR
   AND    CONCEPT.LEARNING
   AND    EXPERIMENTS.IN.INDUCTION
   AND    IPLV.MANUAL
   AND    GPS
   AND    FIGHTS.GAMES.AND.DEBATES
   AND    PANDEMONIUM
   AND    COMPUTING.MACHINERY.AND.INTELLIGENCE
   AND    PATTERN.RECOGNITION

 QUESTION - WHO COAUTHOR.WITH E.HUNT

 WITH THE INFORMATION I NOW HAVE I CANNOT REDEFINE COAUTHOR.WITH(INV)
SO THAT IT CAN BE APPLIED TO E.HUNT.
IF YOU WISH TO ACCEPT THIS AS YOUR ANSWER ENTER - OK
IF YOU WISH TO SUPPLY AN ANSWER, ENTER A STATEMENT OF THE FORM -
 E.HUNT COAUTHOR.WITH(INV) X     OR      COAUTHOR.WITH(INV) EQUIV X
OTHERWISE ENTER A SERIES OF PAIRS X Y SUCH THAT X COAUTHOR.WITH(INV) Y
FOLLOWED BY THE CLASS OF RELATIONS COAUTHOR.WITH(INV) BELONGS TO
FOLLOWED BY AN INTEGER INDICATING THE MAXIMUM NUMBER OF RELATIONS
THAT MAY BE COMPOSED TO GIVE AN ALTERNATE DEFINITION OF COAUTHOR.WITH(INV).
I WILL THEN ATTEMPT TO FIND A MORE SATISFACTORY  ANSWER TO YOUR QUESTION.

 EXAMPLES - J.KATZ     J.FODOR
            A.NEWELL    H.SIMON
            E.FEIGENBAUM    J.FELDMAN
CLASS - BOOK.RELATIONS
DEPTH - 2
 THANK YOU.

 FROM THE EXAMPLES YOU HAVE GIVEN ME, I WOULD GUESS THAT
COAUTHOR.WITH(INV) IS THE SAME AS
      AUTHOR.OF/AUTHOR.OF(INV)/

 I CAN NOT APPLY COAUTHOR.WITH(INV) TO E.HUNT
BUT I KNOW COAUTHOR.WITH(INV) IS THE SAME AS
     AUTHOR.OF/AUTHOR.OF(INV)/

 I FIGURE E.HUNT COAUTHOR.WITH(INV) E.HUNT,J.MARIN,P.STONE,

ANSWER - E.HUNT
   AND    J.MARIN
   AND    P.STONE
 QUESTION - E.FEIGENBAUM WROTE.WHAT.IN 1961
ANSWER - E.FEIGENBAUM AUTHOR.OF
         THE.SIMULATION.OF.VERBAL.LEARNING.BEHAVIOR DATED
         1961

 QUESTION - WHAT WRITTEN.IN 1959,1960,1961,1962,1963

 WITH THE INFORMATION I NOW HAVE I CANNOT REDEFINE WRITTEN.IN(INV)
SO THAT IT CAN BE APPLIED TO 1959.
IF YOU WISH TO ACCEPT THIS AS YOUR ANSWER ENTER - OK
IF YOU WISH TO SUPPLY AN ANSWER, ENTER A STATEMENT OF THE FORM -
 1959 WRITTEN.IN(INV) X     OR      WRITTEN.IN(INV) EQUIV X
OTHERWISE ENTER A SERIES OF PAIRS X Y SUCH THAT X WRITTEN.IN(INV) Y
FOLLOWED BY THE CLASS OF RELATIONS WRITTEN.IN(INV) BELONGS TO
FOLLOWED BY AN INTEGER INDICATING THE MAXIMUM NUMBER OF RELATIONS
THAT MAY BE COMPOSED TO GIVE AN ALTERNATE DEFINITION OF WRITTEN.IN(INV).
I WILL THEN ATTEMPT TO FIND A MORE SATISFACTORY  ANSWER TO YOUR QUESTION.

 STATEMENT - WRITTEN.IN(INV) EQUIV DATED(INV)/
THANK YOU.

 I CAN NOT APPLY WRITTEN.IN(INV) TO 1959
BUT I KNOW WRITTEN.IN(INV) IS THE SAME AS
    DATED(INV)/

 I FIGURE 1959 WRITTEN.IN(INV) PANDEMONIUM,

 I CAN NOT APPLY WRITTEN.IN(INV) TO 1960
BUT I KNOW WRITTEN.IN(INV) IS THE SAME AS
    DATED(INV)/

 I FIGURE 1960 WRITTEN.IN(INV) STEPS.TOWARD.ARTIFICIAL.INTELL
IGENCE,FIGHTS.GAMES.AND.DEBATES,

 I CAN NOT APPLY WRITTEN.IN(INV) TO 1961
BUT I KNOW WRITTEN.IN(INV) IS THE SAME AS
    DATED(INV)/

 I FIGURE 1961 WRITTEN.IN(INV) THE.SIMULATION.OF.VERBAL.LEARN
ING.BEHAVIOR,GPS,PSYCHOLINGUISTICS,

 I CAN NOT APPLY WRITTEN.IN(INV) TO 1962
BUT I KNOW WRITTEN.IN(INV) IS THE SAME AS
    DATED(INV)/

 I FIGURE 1962 WRITTEN.IN(INV) CONCEPT.LEARNING,

 I CAN NOT APPLY WRITTEN.IN(INV) TO 1963
BUT I KNOW WRITTEN.IN(INV) IS THE SAME AS
    DATED(INV)/

 I FIGURE 1963 WRITTEN.IN(INV) HANDBOOK.OF.MATH.PSYCH,COMPUTE
RS.AND.THOUGHT,PATTERN.RECOGNITION,

ANSWER - PANDEMONIUM
```

```
    AND    STEPS.TOWARD.ARTIFICIAL.INTELLIGENCE
    AND    FIGHTS.GAMES.AND.DEBATES
    AND    THE.SIMULATION.OF.VERBAL.LEARNING.BEHAVIOR
    AND    GPS
    AND    PSYCHOLINGUISTICS
    AND    CONCEPT.LEARNING
    AND    HANDBOOK.OF.MATH.PSYCH
    AND    COMPUTERS.AND.THOUGHT
    AND    PATTERN.RECOGNITION

 QUESTION - WHO WRITES.IN THE.STRUCTURE.OF.LANGUAGE

 WITH THE INFORMATION I NOW HAVE I CANNOT REDEFINE WRITES.IN(INV)
SO THAT IT CAN BE APPLIED TO THE.STRUCTURE.OF.LANGUAGE.
IF YOU WISH TO ACCEPT THIS AS YOUR ANSWER ENTER - OK
IF YOU WISH TO SUPPLY AN ANSWER, ENTER A STATEMENT OF THE FORM -
 THE.STRUCTURE.OF.LANGUAGE WRITES.IN(INV) X      OR      WRITES.IN(INV) EQUIV X
OTHERWISE ENTER A SERIES OF PAIRS X Y SUCH THAT X WRITES.IN(INV) Y
FOLLOWED BY THE CLASS OF RELATIONS WRITES.IN(INV) BELONGS TO
FOLLOWED BY AN INTEGER INDICATING THE MAXIMUM NUMBER OF RELATIONS
THAT MAY BE COMPOSED TO GIVE AN ALTERNATE DEFINITION OF WRITES.IN(INV).
I WILL THEN ATTEMPT TO FIND A MORE SATISFACTORY  ANSWER TO YOUR QUESTION.

 STATEMENT - WRITES.IN(INV) EQUIV APPEARED.IN(INV)/AUTHOR.OF(INV)/
THANK YOU.

 I CAN NOT APPLY WRITES.IN(INV) TO THE.STRUCTURE.OF.LANGUAGE
BUT I KNOW WRITES.IN(INV) IS THE SAME AS
     APPEARED.IN(INV)/AUTHOR.OF(INV)/

 I FIGURE THE.STRUCTURE.OF.LANGUAGE WRITES.IN(INV) W.QUINE,Z.
HARRIS,N.CHOMSKY,P.POSTAL,E.KLIMA,M.HALLE,P.ZIFF,J.KATZ,R.CA
RNAP,J.FODOR,

ANSWER - W.QUINE
    AND    Z.HARRIS
    AND    N.CHOMSKY
    AND    P.POSTAL
    AND    E.KLIMA
    AND    M.HALLE
    AND    P.ZIFF
    AND    J.KATZ
    AND    R.CARNAP
    AND    J.FODOR
```

```
THE MEMORY IS ---

WHO = (IS-/1)

/1 = QUESTION,

QUESTION = (IS(INV)-/2)

/2 = WHO,WHAT,WROTE.WHAT.IN,

WHAT = (IS-/3)

/3 = QUESTION,

WROTE.WHAT.IN = (IS-/4)

/4 = QUESTION,

AUTHOR.OF = (MEMBER-/5)

/5 = BOOK.RELATIONS,

BOOK.RELATIONS = (MEMBER(INV)-/6)

/6 = AUTHOR.OF,AUTHOR.OF(INV),PUBLISHED,PUBLISHED(INV),DATE
     D,DATED(INV),REFERENCES,REFERENCES(INV),APPEARED.IN,AP
     PEARED.IN(INV),

AUTHOR.OF(INV) = (MEMBER-/7)

/7 = BOOK.RELATIONS,

PUBLISHED = (MEMBER-/8)

/8 = BOOK.RELATIONS,

PUBLISHED(INV) = (MEMBER-/9)

/9 = BOOK.RELATIONS,

DATED = (MEMBER-/10)

/10 = BOOK.RELATIONS,

DATED(INV) = (MEMBER-/11)

/11 = BOOK.RELATIONS,

REFERENCES = (CLASS-/197)(EQUIV-/196)(MEMBER-/12)

/12 = BOOK.RELATIONS,

REFERENCES(INV) = (MEMBER-/13)

/13 = BOOK.RELATIONS,
APPEARED.IN = ((INV)-/154)(MEMBER-/14)

/14 = BOOK.RELATIONS,

APPEARED.IN(INV) = (MEMBER-/15)

/15 = BOOK.RELATIONS,

E.HUNT = (COAUTHOR.WITH(INV)-/201)(AUTHOR.OF-/16)

/16 = EXPERIMENTS.IN.INDUCTION,CONCEPT.LEARNING,

EXPERIMENTS.IN.INDUCTION = (PUBLISHED.BY-/195)(REFERENCES(I
                           NV)-/155)(REFERENCES-/24)(PUBLIS
                           HED(INV)-/23)(DATED-/20)(AUTHOR.
                           OF(INV)-/17)

/17 = E.HUNT,J.MARIN,P.STONE,

J.MARIN = (AUTHOR.OF-/18)

/18 = EXPERIMENTS.IN.INDUCTION,

P.STONE = (AUTHOR.OF-/19)

/19 = EXPERIMENTS.IN.INDUCTION,

/20 = 1966,

1966 = (DATED(INV)-/21)

/21 = EXPERIMENTS.IN.INDUCTION,

ACADEMIC.PRESS = (PUBLISHED-/22)

/22 = EXPERIMENTS.IN.INDUCTION,

/23 = ACADEMIC.PRESS,

/24 = THE.SIMULATION.OF.VERBAL.LEARNING.BEHAVIOR,PERFORMANC
      E.OF.READING.TASK.BY.AN.EPAM.PROG,CONCEPT.LEARNING,TH
      E.MAGICAL.NUMBER.7+OR-2,STEPS.TOWARD.ARTIFICIAL.INTEL
      LIGENCE,IPLV.MANUAL,GPS,COGNITION.AND.THOUGHT,PANDEMO
      NIUM,COMPUTING.MACHINERY.AND.INTELLIGENCE,

THE.SIMULATION.OF.VERBAL.LEARNING.BEHAVIOR = (-/153)(DATED-
                                             /30)(APPEARED.
                                             IN-/28)(AUTHOR
                                             .OF(INV)-/27)(
                                             REFERENCES(INV
                                             )-/25)

/25 = EXPERIMENTS.IN.INDUCTION,COGNITION.AND.THOUGHT,

E.FEIGENBAUM = (AUTHOR.OF-/26)
```

```
/26 = THE.SIMULATION.OF.VERBAL.LEARNING.BEHAVIOR,PERFORMANC
      E.OF.READING.TASK.BY.AN.EPAM.PROG,COMPUTERS.AND.THOUG
      HT,

/27 = E.FEIGENBAUM,

/28 = PROC.WJCC,

PROC.WJCC = (APPEARED.IN(INV)-/29)

/29 = THE.SIMULATION.OF.VERBAL.LEARNING.BEHAVIOR,

/30 = 1961,

1961 = (WRITTEN.IN(INV)-/206)(DATED(INV)-/31)

/31 = THE.SIMULATION.OF.VERBAL.LEARNING.BEHAVIOR,GPS,PSYCHO
      LINGUISTICS,

PERFORMANCE.OF.READING.TASK.BY.AN.EPAM.PROG = (APPEARED.IN-
                                              /35)(AUTHOR.O
                                              F(INV)-/33)(R
                                              EFERENCES(INV
                                              )-/32)

/32 = EXPERIMENTS.IN.INDUCTION,

/33 = E.FEIGENBAUM,H.SIMON,

H.SIMON = (AUTHOR.OF-/34)

/34 = PERFORMANCE.OF.READING.TASK.BY.AN.EPAM.PROG,GPS,

/35 = BEHAVIORAL.SCI,

BEHAVIORAL.SCI = (APPEARED.IN(INV)-/36)

/36 = PERFORMANCE.OF.READING.TASK.BY.AN.EPAM.PROG,

CONCEPT.LEARNING = (PUBLISHED(INV)-/42)(DATED-/39)(AUTHOR.O
                   F(INV)-/38)(REFERENCES(INV)-/37)

/37 = EXPERIMENTS.IN.INDUCTION,COGNITION.AND.THOUGHT,

/38 = E.HUNT,

/39 = 1962,

1962 = (WRITTEN.IN(INV)-/207)(DATED(INV)-/40)

/40 = CONCEPT.LEARNING,

WILEY = (PUBLISHED-/41)

/41 = CONCEPT.LEARNING,COGNITION.AND.THOUGHT,HANDBOOK.OF.MA
```

```
      TH.PSYCH,

/42 = WILEY,

THE.MAGICAL.NUMBER.7+OR-2 = (APPEARED.IN-/48)(DATED-/46)(AU
                            THOR.OF(INV)-/45)(REFERENCES(IN
                            V)-/43)

/43 = EXPERIMENTS.IN.INDUCTION,

G.MILLER = (AUTHOR.OF-/44)

/44 = THE.MAGICAL.NUMBER.7+OR-2,

/45 = G.MILLER,

/46 = 1956,

1956 = (DATED(INV)-/47)

/47 = THE.MAGICAL.NUMBER.7+OR-2,

/48 = PSYCH.REV.,

PSYCH.REV. = (APPEARED.IN(INV)-/49)

/49 = THE.MAGICAL.NUMBER.7+OR-2,

STEPS.TOWARD.ARTIFICIAL.INTELLIGENCE = (APPEARED.IN-/55)(DA
                                       TED-/53)(AUTHOR.OF(I
                                       NV)-/52)(REFERENCES(
                                       INV)-/50)

/50 = EXPERIMENTS.IN.INDUCTION,

M.MINSKY = (AUTHOR.OF-/51)

/51 = STEPS.TOWARD.ARTIFICIAL.INTELLIGENCE,

/52 = M.MINSKY,

/53 = 1960,

1960 = (WRITTEN.IN(INV)-/205)(DATED(INV)-/54)

/54 = STEPS.TOWARD.ARTIFICIAL.INTELLIGENCE,FIGHTS.GAMES.AND
      .DEBATES,

/55 = PROC.IRE,

PROC.IRE = (APPEARED.IN(INV)-/56)

/56 = STEPS.TOWARD.ARTIFICIAL.INTELLIGENCE,

IPLV.MANUAL = (PUBLISHED(INV)-/63)(DATED-/60)(AUTHOR.OF(INV
              )-/59)(REFERENCES(INV)-/57)
```

```
/57 = EXPERIMENTS.IN.INDUCTION,COGNITION.AND.THOUGHT,

A.NEWELL = (AUTHOR.OF-/58)

/58 = IPLV.MANUAL,GPS,

/59 = A.NEWELL,

/60 = 1964,

1964 = (DATED(INV)-/61)

/61 = IPLV.MANUAL,THE.STRUCTURE.OF.LANGUAGE,

PRENTICE.HALL = (PUBLISHED-/62)

/62 = IPLV.MANUAL,THE.STRUCTURE.OF.LANGUAGE,

/63 = PRENTICE.HALL,

GPS = (APPEARED.IN-/67)(DATED-/66)(AUTHOR.OF(INV)-/65)(REFE
      RENCES(INV)-/64)

/64 = EXPERIMENTS.IN.INDUCTION,COGNITION.AND.THOUGHT,

/65 = A.NEWELL,H.SIMON,

/66 = 1961,

/67 = LERNENDE.AUTOMATEN,

LERNENDE.AUTOMATEN = (PUBLISHED(INV)-/72)(AUTHOR.OF(INV)-/7
                     )(APPEARED.IN(INV)-/68)

/68 = GPS,

H.BILLING = (AUTHOR.OF-/69)

/69 = LERNENDE.AUTOMATEN,

/70 = H.BILLING,

OLDENBOURG = (PUBLISHED-/71)

/71 = LERNENDE.AUTOMATEN,

/72 = OLDENBOURG,

EXPERMENTS.IN.INDUCTION = (REFERENCES-/73)

/73 = FIGHTS.GAMES.AND.DEBATES,

FIGHTS.GAMES.AND.DEBATES = (PUBLISHED(INV)-/79)(DATED-/77)(
                           AUTHOR.OF(INV)-/76)(REFERENCES(I
                           NV)-/74)
```

```
/74 = EXPERMENTS.IN.INDUCTION,COGNITION.AND.THOUGHT,

A.RAPOPORT = (AUTHOR.OF-/75)

/75 = FIGHTS.GAMES.AND.DEBATES,

/76 = A.RAPOPORT,

/77 = 1960,

U.MICHIGAN = (PUBLISHED-/78)

/78 = FIGHTS.GAMES.AND.DEBATES,

/79 = U.MICHIGAN,

COGNITION.AND.THOUGHT = (REFERENCES-/140)(DATED-/138)(PUBLI
                        SHED(INV)-/137)(AUTHOR.OF(INV)-/136
                        )(REFERENCES(INV)-/80)

/80 = EXPERIMENTS.IN.INDUCTION,

PANDEMONIUM = (DATED-/158)(AUTHOR.OF(INV)-/157)(REFERENCES(
              INV)-/81)

/81 = EXPERIMENTS.IN.INDUCTION,COGNITION.AND.THOUGHT,

COMPUTING.MACHINERY.AND.INTELLIGENCE = (APPEARED.IN-/87)(DA
                                       TED-/85)(REFERENCES(
                                       INV)-/82)

/82 = EXPERIMENTS.IN.INDUCTION,COGNITION.AND.THOUGHT,

A.TURING = (AUTHOR.OF-/83)

/83 = COMPUTING.MACHINES.AND.INTELLIGENCE,

COMPUTING.MACHINES.AND.INTELLIGENCE = (AUTHOR.OF(INV)-/84)

/84 = A.TURING,

/85 = 1950,

1950 = (DATED(INV)-/86)

/86 = COMPUTING.MACHINERY.AND.INTELLIGENCE,

/87 = MIND,

MIND = (APPEARED.IN(INV)-/88)

/88 = COMPUTING.MACHINERY.AND.INTELLIGENCE,

J.KATZ = (AUTHOR.OF-/89)
```

```
/89 = THE.STRUCTURE.OF.LANGUAGE,SEMISENTENCES,THE.STRUCTURE
      .OF.A.SEMANTIC.THEORY,

THE.STRUCTURE.OF.LANGUAGE = (WRITES.IN(INV)-/211)(APPEARED.
                            IN(INV)-/87)(DATED-/93)(PUBLISH
                            ED(INV)-/92)(AUTHOR.OF(INV)-/90
                            )

/90 = J.KATZ,J.FODOR,

J.FODOR = (AUTHOR.OF-/91)

/91 = THE.STRUCTURE.OF.LANGUAGE,THE.STRUCTURE.OF.A.SEMANTIC
      .THEORY,

/92 = PRENTICE.HALL,

/93 = 1964,

W.QUINE = (AUTHOR.OF-/94)

/94 = THE.PROBLEM.OF.MEANING.IN.LINGUISTICS,SPEAKING.OF.OBJ
      ECTS,

THE.PROBLEM.OF.MEANING.IN.LINGUISTICS = (APPEARED.IN-/96)(A
                                        UTHOR.OF(INV)-/95)

/95 = W.QUINE,

/96 = THE.STRUCTURE.OF.LANGUAGE,

/97 = THE.PROBLEM.OF.MEANING.IN.LINGUISTICS,DISTRIBUTIONAL.
      STRUCTURE,CURRENT.ISSUES.IN.LINGUISTIC.THEORY,ON.THE.
      NOTION.RULE.OF.GRAMMAR,LIMITATIONS.OF.PHRASE.STRUCTUR
      E.GRAMMARS,A.TRANSFORMATIONAL.APPROACH.TO.SYNTAX,NEGA
      TION.IN.ENGLISH,ON.THE.BASES.OF.PHONOLOGY,PHONOLOGY.I
      N.GENERATIVE.GRAMMAR,DISCOURSE.ANALYSIS,DEGREES.OF.GR
      AMMATICALNESS,ON.UNDERSTANDING.UNDERSTANDING.UTTERANC
      ES,SEMISENTENCES,FOUNDATIONS.OF.LOGIC.AND.MATHEMATICS
      ,SPEAKING.OF.OBJECTS,THE.STRUCTURE.OF.A.SEMANTIC.THEO
      RY,

Z.HARRIS = (AUTHOR.OF-/98)

/98 = DISTRIBUTIONAL.STRUCTURE,DISCOURSE.ANALYSIS,

DISTRIBUTIONAL.STRUCTURE = (APPEARED.IN-/100)(AUTHOR.OF(INV
                           )-/99)

/99 = Z.HARRIS,

/100 = THE.STRUCTURE.OF.LANGUAGE,

N.CHOMSKY = (AUTHOR.OF-/101)

/101 = CURRENT.ISSUES.IN.LINGUISTIC.THEORY,ON.THE.NOTION.RU
```

```
          LE.OF.GRAMMAR,A.TRANSFORMATIONAL.APPROACH.TO.SYNTAX,
          DEGREES.OF.GRAMMATICALNESS,THE.INDEPENDENCE.OF.GRAMM
          AR,

CURRENT.ISSUES.IN.LINGUISTIC.THEORY = (APPEARED.IN-/103)(AU
                                      THOR.OF(INV)-/102)

/102 = N.CHOMSKY,

/103 = THE.STRUCTURE.OF.LANGUAGE,

ON.THE.NOTION.RULE.OF.GRAMMAR = (APPEARED.IN-/105)(AUTHOR.O
                                F(INV)-/104)

/104 = N.CHOMSKY,

/105 = THE.STRUCTURE.OF.LANGUAGE,

P.POSTAL = (AUTHOR.OF-/106)

/106 = LIMITATIONS.OF.PHRASE.STRUCTURE.GRAMMARS,

LIMITATIONS.OF.PHRASE.STRUCTURE.GRAMMARS = (APPEARED.IN-/10
                                           8)(AUTHOR.OF(INV
                                           )-/107)

/107 = P.POSTAL,

/108 = THE.STRUCTURE.OF.LANGUAGE,

A.TRANSFORMATIONAL.APPROACH.TO.SYNTAX = (APPEARED.IN-/110)(
                                        AUTHOR.OF(INV)-/109
                                        )

/109 = N.CHOMSKY,

/110 = THE.STRUCTURE.OF.LANGUAGE,

E.KLIMA = (AUTHOR.OF-/111)

/111 = NEGATION.IN.ENGLISH,

NEGATION.IN.ENGLISH = (APPEARED.IN-/113)(AUTHOR.OF(INV)-/11
                      2)

/112 = E.KLIMA,

/113 = THE.STRUCTURE.OF.LANGUAGE,

M.HALLE = (AUTHOR.OF-/114)

/114 = ON.THE.BASES.OF.PHONOLOGY,PHONOLOGY.IN.GENERATIVE.GR
          AMMAR,PHONEMIC.PATTERNING,

ON.THE.BASES.OF.PHONOLOGY = (APPEARED.IN-/116)(AUTHOR.OF(IN
                            V)-/115)
```

```
/115 = M.HALLE,

/116 = THE.STRUCTURE.OF.LANGUAGE,

PHONOLOGY.IN.GENERATIVE.GRAMMAR = (APPEARED.IN-/118)(AUTHOR
                                  .OF(INV)-/117)

/117 = M.HALLE,

/118 = THE.STRUCTURE.OF.LANGUAGE,

DISCOURSE.ANALYSIS = (APPEARED.IN-/120)(AUTHOR.OF(INV)-/119
                     )

/119 = Z.HARRIS,

/120 = THE.STRUCTURE.OF.LANGUAGE,

DEGREES.OF.GRAMMATICALNESS = (APPEARED.IN-/122)(AUTHOR.OF(I
                             NV)-/121)

/121 = N.CHOMSKY,

/122 = THE.STRUCTURE.OF.LANGUAGE,

P.ZIFF = (AUTHOR.OF-/123)

/123 = ON.UNDERSTANDING.UNDERSTANDING.UTTERANCES,

ON.UNDERSTANDING.UNDERSTANDING.UTTERANCES = (APPEARED.IN-/1
                                            25)(AUTHOR.OF(I
                                            NV)-/124)

/124 = P.ZIFF,

/125 = THE.STRUCTURE.OF.LANGUAGE,

SEMISENTENCES = (APPEARED.IN-/127)(AUTHOR.OF(INV)-/126)

/126 = J.KATZ,

/127 = THE.STRUCTURE.OF.LANGUAGE,

R.CARNAP = (AUTHOR.OF-/128)

/128 = FOUNDATIONS.OF.LOGIC.AND.MATHEMATICS,

FOUNDATIONS.OF.LOGIC.AND.MATHEMATICS = (APPEARED.IN-/130)(A
                                       UTHOR.OF(INV)-/129)

/129 = R.CARNAP,

/130 = THE.STRUCTURE.OF.LANGUAGE,

SPEAKING.OF.OBJECTS = (APPEARED.IN-/132)(AUTHOR.OF(INV)-/13
                         1)

/131 = W.QUINE,

/132 = THE.STRUCTURE.OF.LANGUAGE,

THE.STRUCTURE.OF.A.SEMANTIC.THEORY = (APPEARED.IN-/134)(AUT
                                     HOR.OF(INV)-/133)

/133 = J.KATZ,J.FODOR,

/134 = THE.STRUCTURE.OF.LANGUAGE,

W.REITMAN = (REFERENCES-/198)(AUTHOR.OF-/135)

/135 = COGNITION.AND.THOUGHT,

/136 = W.REITMAN,

/137 = WILEY,

/138 = 1965,

1965 = (DATED(INV)-/139)

/139 = COGNITION.AND.THOUGHT,

/140 = INTR.TO.THE.FORMAL.ANAL.OF.NATURAL.LANGUAGES,COMPUTE
       RS.AND.THOUGHT,THE.SIMULATION.OF.VERBAL.LEARNING.BEH
       AVIOR,CONCEPT.LEARNING,EXPERIMENTS.IN.INDUCTION,IPLV
       .MANUAL,GPS,EIGHTS.GAMES.AND.DEBATES,PANDEMONIUM,COM
       PUTING.MACHINERY.AND.INTELLIGENCE,PATTERN.RECOGNITIO
       N,

INTR.TO.THE.FORMAL.ANAL.OF.NATURAL.LANGUAGES = (AUTHOR.OF(I
                                               NV)-/192)(AP
                                               PEARED.IN-/1
                                               42)(REFERENC
                                               ES(INV)-/141
                                               )

/141 = COGNITION.AND.THOUGHT,

/142 = HANDBOOK.OF.MATH.PSYCH,

HANDBOOK.OF.MATH.PSYCH = (DATED-/145)(PUBLISHED(INV)-/144)(
                         APPEARED.IN(INV)-/143)

/143 = INTR.TO.THE.FORMAL.ANAL.OF.NATURAL.LANGUAGES,

/144 = WILEY,

/145 = 1963,

1963 = (WRITTEN.IN(INV)-/208)(DATED(INV)-/146)
```

```
/146  =  HANDBOOK.OF.MATH.PSYCH,COMPUTERS.AND.THOUGHT,PATTERN
         .RECOGNITION,

COMPUTERS.AND.THOUGHT  =  (DATED-/152)(PUBLISHED(INV)-/151)(A
                           UTHOR.OF(INV)-/148)(REFERENCES(INV)
                            /147)

/147  =  COGNITION.AND.THOUGHT,

/148  =  E.FEIGENBAUM,J.FELDMAN,

J.FELDMAN  =  (AUTHOR.OF-/149)

/149  =  COMPUTERS.AND.THOUGHT,

MCGRAW.HILL  =  (PUBLISHED-/150)

/150  =  COMPUTERS.AND.THOUGHT,

/151  =  MCGRAW.HILL,

/152  =  1963,

/153  =  APPEARED.IN,

/154  =  THE.SIMULATION.OF.VERBAL.LEARNING.BEHAVIOR,

/155  =  COGNITION.AND.THOUGHT,

O.SELFRIDGE  =  (AUTHOR.OF-/156)

/156  =  PANDEMONIUM,

/157  =  O.SELFRIDGE,

/158  =  1959,

1959  =  (WRITTEN.IN(INV)-/204)(DATED(INV)-/159)

/159  =  PANDEMONIUM,

PATTERN.RECOGNITION  =  (DATED-/165)(APPEARED.IN-/163)(AUTHOR
                         .OF(INV)-/162)(REFERENCES(INV)-/160)

/160  =  COGNITION.AND.THOUGHT,

L.UHR  =  (AUTHOR.OF-/161)

/161  =  PATTERN.RECOGNITION,

/162  =  L.UHR,

/163  =  PSYCHOL.BULL.

PSYCHOL.BULL  =  (APPEARED.IN(INV)-/164)
```

```
/164 = PATTERN.RECOGNITION,

/165 = 1963,

PSYCHOLINGUISTICS = (APPEARED.IN(INV)-/172)(PUBLISHED(INV)-
                    /170)(AUTHOR.OF(INV)-/168)(DATED-/166)

/166 = 1961,

S.SAPORTA = (AUTHOR.OF-/167)

/167 = PSYCHOLINGUISTICS,

/168 = S.SAPORTA,

HOLT.RINEHART.AND.WINSTOR = (PUBLISHED-/169)

/169 = PSYCHOLINGUISTICS,

/170 = HOLT.RINEHART.AND.WINSTOR,

LINGUISTICS.SYMBOLS.MAKE.MAN = (AUTHOR.OF(INV)-/174)(APPEAR
                               ED.IN-/171)

/171 = PSYCHOLINGUISTICS,

/172 = LINGUISTICS.SYMBOLS.MAKE.MAN,ON.LINGUISTIC.TERMS,THE
       .INDEPENDENCE.OF.GRAMMAR,MEANING,LANGUAGE.DEVELOPMEN
       T.IN.CHILDREN,PHONEMIC.PATTERNING,APHASIA.AS.A.LINGU
       ISTIC.PROBLEM,

J.LOTZ = (AUTHOR.OF-/173)

/173 = LINGUISTICS.SYMBOLS.MAKE.MAN,

/174 = J.LOTZ,

ON.LINGUISTIC.TERMS = (AUTHOR.OF(INV)-/177)(APPEARED.IN-/17
                      5)

/175 = PSYCHOLINGUISTICS,

F.HOUSEHOLDER = (AUTHOR.OF-/176)

/176 = ON.LINGUISTIC.TERMS,

/177 = F.HOUSEHOLDER,

THE.INDEPENDENCE.OF.GRAMMAR = (AUTHOR.OF(INV)-/179)(APPEARE
                              D.IN-/178)

/178 = PSYCHOLINGUISTICS,

/179 = N.CHOMSKY,

MEANING = (AUTHOR.OF(INV)-/182)(APPEARED.IN-/180)
```

```
/180 = PSYCHOLINGUISTICS,

L.BLOOMFIELD = (AUTHOR.OF-/181)

/181 = MEANING,

/182 = L.BLOOMFIELD,

LANGUAGE.DEVELOPMENT.IN.CHILDREN = (AUTHOR.OF(INV)-/185)(AP
                                    PEARED.IN-/183)

/183 = PSYCHOLINGUISTICS,

J.CARROLL = (AUTHOR.OF-/184)

/184 = LANGUAGE.DEVELOPMENT.IN.CHILDREN,

/185 = J.CARROLL,

PHONEMIC.PATTERNING = (AUTHOR.OF(INV)-/188)(APPEARED.IN-/18
                      6)

/186 = PSYCHOLINGUISTICS,

R.JAKOBSON = (AUTHOR.OF-/187)

/187 = PHONEMIC.PATTERNING,APHASIA.AS.A.LINGUISTIC.PROBLEM,

/188 = R.JAKOBSON,M.HALLE,

APHASIA.AS.A.LINGUISTIC.PROBLEM = (AUTHOR.OF(INV)-/190)(APP
                                  EARED.IN-/189)

/189 = PSYCHOLINGUISTICS,

/190 = R.JAKOBSON,

CHOMSKY = (AUTHOR.OF-/191)

/191 = INTR.TO.THE.FORMAL.ANAL.OF.NATURAL.LANGUAGES,

/192 = CHOMSKY,

PUBLISHED.BY = (CLASS-/194)(EQUIV-/193)

/193 = PUBLISHED(INV)/,

/194 = PUBLISHED(INV),

/195 = ACADEMIC.PRESS,

/196 = AUTHOR.OF/REFERENCES/,

/197 = AUTHOR.OF,
/198  =  INTR.TO.THE.FORMAL.ANAL.OF.NATURAL.LANGUAGES,COMPUTE
         RS.AND.THOUGHT,THE.SIMULATION.OF.VERBAL.LEARNING.BEH
         AVIOR,CONCEPT.LEARNING,EXPERIMENTS.IN.INDUCTION,IPLV
         .MANUAL,GPS,FIGHTS.GAMES.AND.DEBATES,PANDEMONIUM,COM
         PUTING.MACHINERY.AND.INTELLIGENCE,PATTERN.RECOGNITIO
         N,

COAUTHOR.WITH(INV)  =  (CLASS-/200)(EQUIV-/199)

/199  =  AUTHOR.OF/AUTHOR.OF(INV)/,

/200  =  AUTHOR.OF,

/201  =  E.HUNT,J.MARIN,P.STONE,

WRITTEN.IN(INV)  =  (CLASS-/203)(EQUIV-/202)

/202  =  DATED(INV)/,

/203  =  DATED(INV),

/204  =  PANDEMONIUM,

/205  =  STEPS.TOWARD.ARTIFICIAL.INTELLIGENCE,FIGHTS.GAMES.AN
         D.DEBATES,

/206  =  THE.SIMULATION.OF.VERBAL.LEARNING.BEHAVIOR,GPS,PSYCH
         OLINGUISTICS,

/207  =  CONCEPT.LEARNING,

/208  =  HANDBOOK.OF.MATH.PSYCH,COMPUTERS.AND.THOUGHT,PATTERN
         .RECOGNITION,

WRITES.IN(INV)  =  (CLASS-/210)(EQUIV-/209)

/209  =  APPEARED.IN(INV)/AUTHOR.OF(INV)/,

/210  =  APPEARED.IN(INV),

/211  =  W.QUINE,Z.HARRIS,N.CHOMSKY,P.POSTAL,E.KLIMA,M.HALLE,
         P.ZIFF, J.KATZ,R.CARNAP,J.FODOR,
```