A SYNTAXLESS QUESTION-ANSWERING SYSTEM

Antonio Leal
November 23, 1973

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

The motivation for this project stems from my belief that a computer question-answering system could be built that depends solely on a keyword search scheme for understanding questions. Further-- that it would operate within satisfactory limits so that it would actually be useful. It was my intention to purposefully use as little syntactic or structural information as possible to find out just how much could be done with purely a semantic approach. Hopefully, it will lead to ideas on how to incorporate syntax analysis within its structure to expand its power. The system was written in LISP and the description that follows uses a model-theoretic notation.

II. THE MODEL

Let C be the set of constant symbols:

$$C=\{o,p_1,p_2,w_1,w_2,c_1,c_2,s,d,e_1,e_2\}$$

where each symbol denotes a set. Let O be the set of objects in the data base. Then,

$o=O\cup\{$*certain subsets of* $O\}\cup\{O\}\cup\{$they,them,their,it,its, one,ones,other,others,these,those$\}$

$p_1=\{$*properties*$\}$

$p_2=\{$*certain subsets of* $p_1\}\cup\{p_1\}$

$w_1=\{$what,which,who,tell,print,give$\}$

$w_2=\{$many$\}$

$c_1=\{$and,&,but,except,neither$\}$

$c_2=\{$or$\}$

$s=\{$*stoppers*$\}$

$d=\{$*general words referring to the data base as a whole*$\}$

$e_1=\{$all$\}$

$e_2=\{$any,some$\}$

Every member of each of the sets must be distinct and unique from all of the others. Given with C is a function

$$f: O \times p_1 \rightarrow \{\text{yes, no, I don't know}\}$$

that relates objects and properties. The set o contains all of the names of the objects that are the subject of the data base. If certain subsets of objects are naturally thought of as a group, the words naming the subsets are added to o. Also in o are words referring to the entire set of objects as well as anaphoric words such as "they", "it", etc. In general, the words in o serve to establish a topic of discussion. The objects are the center of attention and they may be referred to explicitly, as a whole, as members of a group, or by anaphora. For this discussion, it is not important whether o contains the names of the objects or the words representing the names of the objects. We will refer to o in both senses and, sometimes, as if it contained the objects themselves.

The properties are divided into two sets since it is more important to know what type of property words are being used in the question. p_1 contains all of the specific property words and p_2 contains the words denoting subsets of p_1 plus global property words. The set s contains words that, for some reason or another, signal that the question cannot be answered. s would contain items like numbers, special characters, and words close to the domain of discourse but, for which, no information is available in the data base. The set d contains words referring to the data base as a whole that are independent of the data.

Let Q be the set of recognizable words in the question and let P be a monadic predicate symbol. Then Φ_Q is a set of interpretations such that:

$$P(x)=T \leftrightarrow x \cap Q \neq \phi$$
$$P(x)=F \leftrightarrow x \cap Q = \phi$$

That is, P(x) is true if and only if at least one of the words in x appears in the question. We are interested in the following set of clauses:

$$S=\{\sim P(s), P(o) \vee P(p_1) \vee P(p_2) \vee P(d), \sim P(w_1) \vee \sim P(w_2), \sim P(e_1) \vee \sim P(e_2), \sim P(c_1) \vee \sim P(c_2)\}$$

and the associated finite semantic tree. For each Φ_Q an answer $\alpha(\Phi_Q)$ will be provided. Figure 1 shows the top of the semantic tree. The tips are labeled τ_i and figure 2 shows a representative subtree for τ_i $1 \leq i \leq 7$. These subtrees are not complete. $P(c_1)$ and $P(c_2)$ are left out because they have no significance in determining α. P(d) only appears in τ_8 shown in figure 3.

τ_1: The τ_1 branch of the semantic tree has $P(o)=P(p_1)=P(p_2)=T$. This means that there are not only specific objects mentioned in the question, but also specific properties and property set words. Thus,

$$\alpha(\Phi_Q:\tau_1)=\{x \epsilon p_2 \cap Q / f(f(o \cap Q, p_1 \cap Q)=yes, x)=yes\}$$

That is, those properties from the subset mentioned in the question that apply to the objects mentioned qualified by the specific properties. $\alpha(\Phi_Q:\tau_i)$ is α applied to the partial interpretation up to τ_i and is considered to be the same for all Φ_Q containing $\Phi_Q:\tau_i$. Although α is the answer to the question, the actual answer that is printed on the user's terminal is determined by the full Φ_Q, i.e. the complete path.

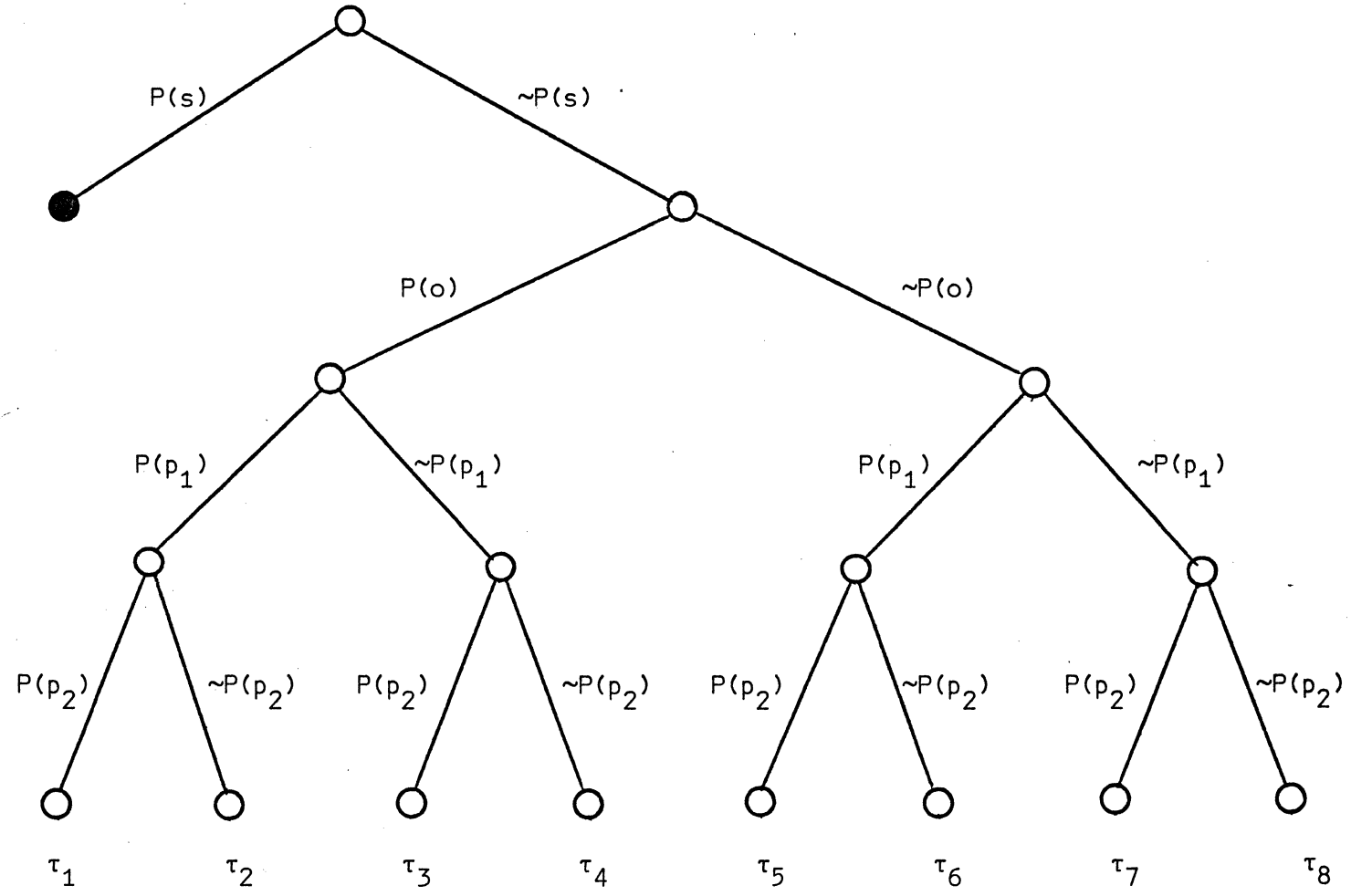

Figure 1: Top of the Semantic Tree

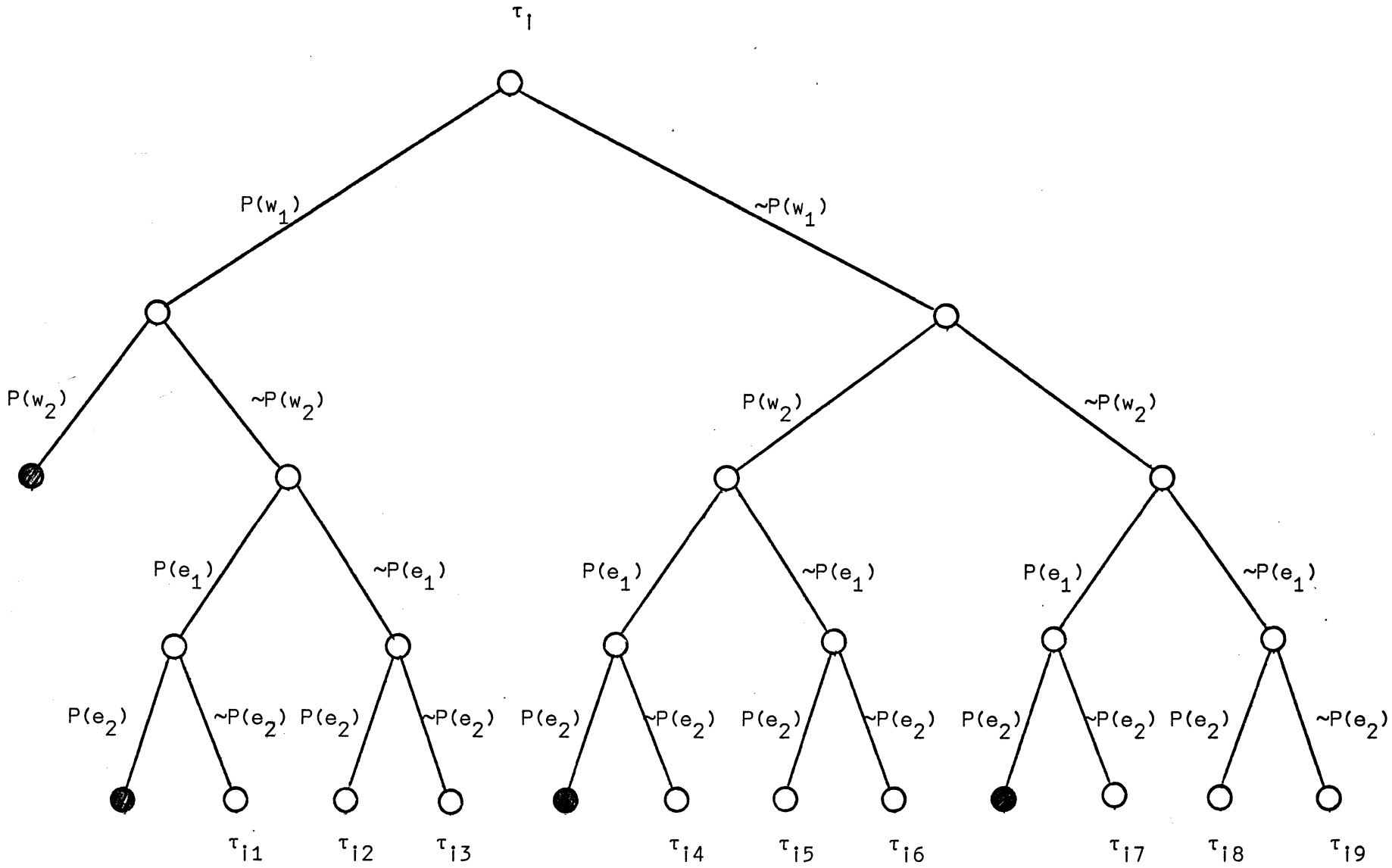

Figure 2: The τ_i subtree

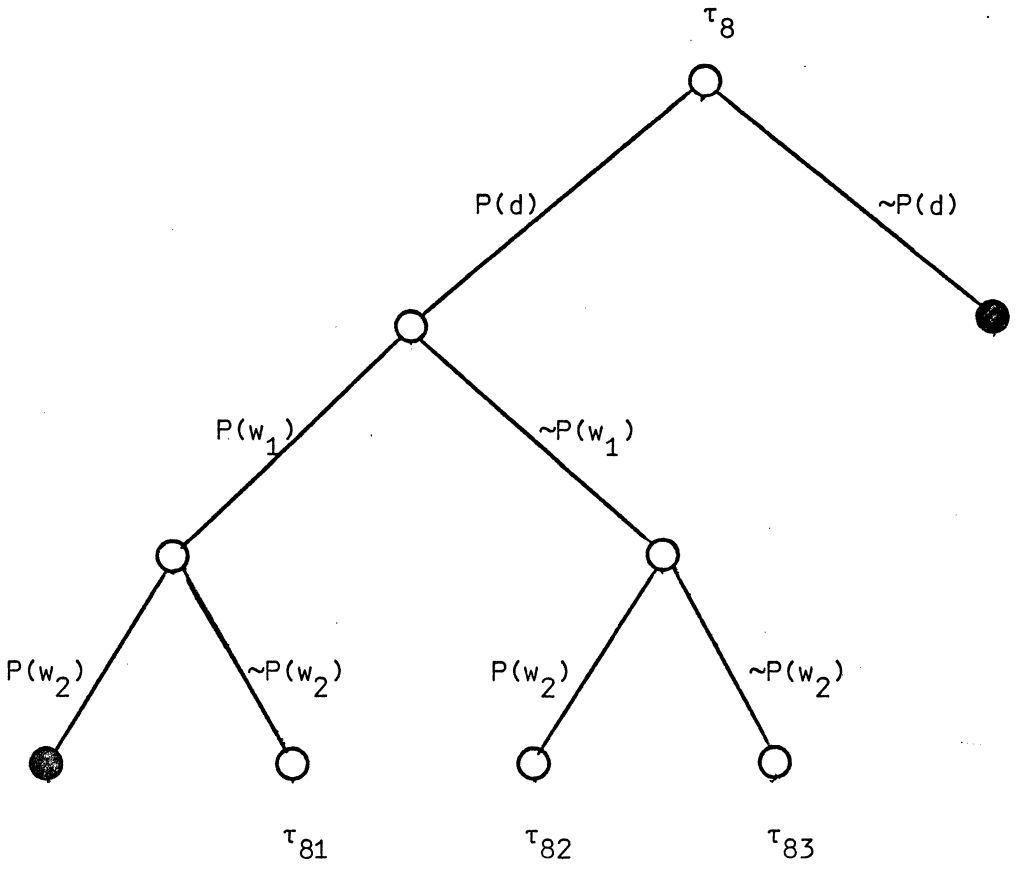

Figure 3: The τ_8 subtree

For example, τ_{i1} asks, "What are all...?"; τ_{i6} asks, "How many...?"; and τ_{i8} asks, "Are there any...?" Thus, α is the constructed set of answers that may or may not be printed.

τ_2: The questions in the τ_2 branch just contain objects and specific properties.

$$\alpha(\Phi_Q:\tau_2)=\{x\varepsilon on Q/f(x,p_1 n Q)=yes\}$$

Thus, it is a simple question of whether or not specific properties apply to specific objects. For example,

τ_{23}: What objects have X? $X\varepsilon p_1$

τ_{26}: How many X and Y objects are there? $X,Y\varepsilon p_1$

τ_3: In the τ_3 subtree, specific objects are found in the question along with property group names.

$$\alpha(\Phi_Q:\tau_3)=\{x\varepsilon p_2 n Q/f(on Q,x)=yes\}$$

For example,

τ_{31}: What properties in such and such group does that object have?

τ_{37}: Do all objects have properties in this particular group?

τ_4: The τ_4 subtree implies that only objects appear in the question. In this case, the object names are simply printed out. τ_4 questions very often arise from anaphoric reference.

$$\alpha(\Phi_Q:\tau_4)=o\cap Q$$

For example,

τ_{42}: What are some objects?

τ_{43}: What are they? What is it?

τ_{46}: How many of them are there? How many objects are there?

τ_{49}: Is X an object?

The τ_5 through τ_8 questions have no objects in them. Thus, they can only refer to information about the properties. This makes them somewhat easier to handle but introduces some ambiguities.

τ_5: Both a specific property and a property group occur in Q.

$$\alpha(\Phi_Q:\tau_5)=\{p\varepsilon p_1\cap Q/p\varepsilon p_2\cap Q\}$$

Most of the branches have no meaning.

τ_{59}: Is this property a member of this group?

τ_6: All of the τ_6 questions are meaningless. The only thing in the question that is recognizable is one or more specific property names.

$$\alpha(\Phi_Q:\tau_6)=K_1$$

where K_1 is the constant answer "a property." For example,

τ_{63}: What is X? $X \varepsilon p_1$

τ_7: The τ_7 questions have a little more substance than the τ_6 ones. We have the name of at least one property group to give information about.

$$\alpha(\Phi_Q:\tau_7)=p_2 \cap Q$$

For example,

τ_{71}: What are all of the properties of this group?

τ_{76}: How many properties of this type are there?

τ_{79}: Is X a member of this group? $X \not\varepsilon p_2$

We must be careful about τ_{79}. Since the property group name is all that is recognizable in Q, the word X (see above example) is unknown. If we assume that it is present, the answer should be "no". However, it is impossible to tell the difference between the example question given above and:

τ_{79}: Are there properties?

for which we want to answer "yes". Since the former question is more specific and more likely to be asked, τ_{79} questions get "no" as an answer.

τ_8: In τ_8 questions, no objects, no properties, and no property groups occur in Q. (see Figure 3) If it also happened that ~P(d), there would nothing recognizable in the question for which an answer could be given. Thus the reason for the clause in S. If we have P(d), a general question has been asked, and it is enough to print out a canned information message.

$$\alpha(\Phi_Q:\tau_8)=K_2$$

For example,

τ_{81}: What is in the data base?

τ_{81}: Give me some information.

Anaphoric reference words are handled in a special way. The set o has specific object words as well as anaphoric reference words. In order to explain the mechanism, o has to be split up. Let:

a_1={they,them,their,it,its,one,ones,other,others}
a_2={these,those}
$o'=o-(a_1 \cup a_2)$

A question Q is called a *root* question if

$$o' \cap Q \neq \phi$$

Thus, any question that has specific objects or object groups occurring in it is a root question. Such questions establish $D \subseteq O$, the set of objects under current consideration. Such a subset D is called the current domain of discourse.

$$o' \cap Q \neq \phi \rightarrow D = \alpha(\Phi_Q)$$

$$o' \cap Q = \phi \rightarrow D = \phi$$

D does not change with subsequent Q's unless (1) a new domain is established with a new root question, (2) a question is asked that does not contain any objects at all, or (3) a new domain is established through the use of an a_2 word. If, on subsequent questions,

$$a_1 \cap Q \neq \phi$$

we form a sub-domain $D' \subseteq D$ which contains the objects of D possibly further qualified.

[1] $(a_1 \cap Q \neq \phi \wedge D = \phi) \rightarrow \alpha(\Phi_Q) = \text{"?"}$

[2] $(a_1 \cap Q \neq \phi \wedge D \neq \phi) \rightarrow D' = \alpha(\Phi_Q)$

[3] $o' \cap Q = \phi \rightarrow D' = \phi$

In [1], anaphoric reference was used in a question where no previous D had been established. Such a question has no meaning. In [2], $D' \subseteq D$ is established. Subsequent questions containing a_1 words always refer back to the previous D. D' will change on each Q. Questions containing a_2 words refer back to the D' domain.

[4] $(a_2 \cap Q \neq \phi \wedge D = \phi) \rightarrow \alpha(\Phi_Q) = \text{"?"}$

[5] $(a_2 \cap Q \neq \phi \wedge D \neq \phi \wedge D' \neq \phi) \rightarrow (D = \alpha(\Phi_Q) \wedge D' = \phi)$

[6] $(a_2 \cap Q \neq \phi \wedge D \neq \phi \wedge D' = \phi) \rightarrow D' = \alpha(\Phi_Q)$

In [4], as previously in [1], an a_2 word was used with no established domain. In [6], D' changes but D does not. In [5], an a_2 word causes a new domain D to be established. An example of a sequence of anaphoric questions is given in the next section on the implemented data base.

The treatment of "and" and "or" was more difficult. Without syntax, it is not possible to tell where the connectives appear in the question. This is the reason for the clause:

$$\sim P(c_1) \vee \sim P(c_2)$$

If no connectives appear, "and" is assumed. The situation is made somewhat easier by the fact that if either a single object or a single property appear in the question, it is obvious what the connective applies to. The only bad case is when multiple objects and properties occur in the question along with "or".

It has been assumed that negation rules were applied before the proper interpretation was determined. Negation is the only place where context sensitive rules are applied. In general, when the word "not" appears in the question, the next recognizable object or property is negated. Two other types of negation that are handled are "except" and "neither".

except X → all O and not X
neither X nor Y → not X and not Y

This works fairly well in most cases.

The word "all" is recognized only as an indicator that the user wishes to see the entire list of answers printed. If "all" is not in the question, only 20 of the answers are printed out after which the

user is asked if he wants to see another 20. If the word "some" or "any" is used, only a few answers are printed.

III. IMPLEMENTATION

The program that implements the semantic tree is written in LISP. The objects are names of computer programs and the properties are descriptive words that apply to the programs. Appendix B is a complete list of the descriptive words. Included as properties are such words as "analysis", "chi-square", "probability", Fortran", "IBM360", "UNIVAC", etc. There are no subsets of objects and the word "program" refers to the whole set. The property groups are "language", "computer", and "manufacturer". The words "property" and "keyword" refer to the whole set. The stopper words are

s={fast,documented,system,systems,redistributable,management, modularized,where,topic,topics,documentation,source, distributor,example,core,big,subroutine,more,greater,less, only,percent,this,+,#,$,*,:,!,@,;,/,%,),(,[,],=,"}u{*digits*}

and d={database,data,memory,information,know,knowledge}

Here are some sample questions from the implemented data base for some of the more interesting τ's. Recognized words are underlined.

τ_{13}: What programs are written in languages other than Fortran?
τ_{13}: What computers do analysis programs run on?
τ_{21}: Print all of the Algol programs.
τ_{23}: What Fortran programs deal with matrix inversion?
τ_{26}: How many programs are written in PL/I but not in Algol?
τ_{28}: Are there any UNIVAC programs that are compatible with IBM?
τ_{29}: Does the Burroughs5500 accept programs not written in Algol?

τ_{33}: What language is EDIT written in?
τ_{38}: Do you know if EDIT runs on any specific computer?
τ_{41}: List all of the programs.
τ_{42}: What are some of them?
τ_{43}: What are they?
τ_{43}: What is it?
τ_{46}: How many of them are there?
τ_{59}: Is Fortran a language?
τ_{59}: Is IBM360 among the computers?
τ_{59}: Is inversion a keyword?
τ_{63}: What is analysis?
τ_{71}: What are all of the languages?
τ_{72}: Give me some computers.
τ_{76}: How many manufacturers are there?
τ_{76}: Are there many computers?
τ_{79}: Is Cobol a language?
τ_{81}: What is in the database?
τ_{81}: What do you know?
τ_{81}: Give me some information.
τ_{82}: How many things are there in your database?

Here are some sequential anaphoric reference examples:

[1] How many languages are there?
(no reference; $D=\phi$; $D'=\phi$)

[2] How many programs are written in PL/I?
(root; no reference; $D=\alpha(\Phi_{[2]})$; $D'=\phi$)

[3] How many of them run on the CDC6600?
(refers to [2]; $D=\alpha(\Phi_{[2]})$; $D'=\alpha(\Phi_{[3]})$)

[4] How many of them come from IBM?
(refers to [2];$D=\alpha(\Phi_{[2]})$;$D'=\alpha(\Phi_{[4]})$)

[5] Which of those deal with probability?
(refers to [4];$D=\alpha(\Phi_{[5]})$;$D'=\phi$)

[6] How many of them use covariance calculation?
(refers to [5];$D=\alpha(\Phi_{[5]})$;$D'=\alpha(\Phi_{[6]})$)

[7] What are these called?
(refers to [6];$D=\alpha(\Phi_{[6]})$;$D'=\phi$)

IV. GENERAL REMARKS

Actually, I am surprized that so many different types of questions could be recognized with so little syntax analysis. It is interesting that in playing with the system myself, I tend to ask questions in normal English even though I know which words are being recognized and which are not. In such a simple non-numeric world, complicated questions don't seem to come up. In testing the system out on others, I have found that a little explanation of what is in the data base goes a long way. Someone who tries to use it knowing nothing except that it answers questions about computer programs has great difficulty getting started. However, once started, almost every question gets an answer. Although it is general for any non-hierarchical non-numeric data base of objects and properties, that is still quite a restricted world. I believe that I have gone just about as far as I can without using syntax analysis. I believe, though, that syntax analysis should only be used when semantic analysis cannot produce a meaning. I also believe that a very good English understanding system can be built using the techniques in this paper as a base. Possible extension areas are:

(1) Allowing numeric data in which operations such as counting, comparisons, statistics, and arithmetic operations are allowed.

(2) Allowing object and property structures. For example, a corporate employee data base where employees are in projects which are in branches which are in divisions, etc.

(3) Expanded "and"/"or" capability.

(4) Allowing a single word to have more than one meaning. The meaning would be found from context.

(5) Allowing ideoms, that is, two or more words which mean a single thing when found in a specific order. For example, data management system.

(6) Multiple questions connected with "and" should all be answered.

(7) Spelling correction.

(8) Morphology (word endings such as plurals)

There is no reason why data cannot be entered or changed in the current system although it was not implemented. The same technique can be used effectively. For example, sentences such as:

EDIT is written in Algol.
"Cobol" is a language.

The currently implemented data base is "closed". That is,

$$(\forall o \varepsilon O)(\forall\ p \varepsilon p_1)\quad f(o,p) \neq \text{I don't know}$$

If the data base were not closed, the entry of new information could be handled in the following way. If new data is entered that was not there previously, it is given the value specified in the sentence. However, if the information in the sentence contradicts data already stored, a value of "I don't know" could be given. Thus, the system would have to be told something twice in order to become a fact.

The data comes from a tape made at the University of Wisconsin of computer programs from the social sciences and associated keywords. There are approximately 500 programs and 1100 keywords. The program itself uses 85 pages (at 1024 words per page) of core memory. Of that, approximately 48 pages comprise the LISP compiler, 30 pages are taken up by data, 5 pages by the program, and 12 are left as working space. The response time is from 5-8 seconds on the average. This is very fast compared to syntax analysis programs but the comparison is not really fair since only simple sentences are handled. A question with a lot of negations may take up to 30 seconds since the entire data base must be searched. In such cases, a polite message is printed when the waiting time exceeds a certain limit.

APPENDIX 'I

DEMONSTRATION PRINTOUT

```
List the properties.
     ACCESS ACHIEVED ADDITIVE-MODEL ADJACENT ADJUSTED AGE AGGREGATED
AGREEMENT AIKEN ALGEBRAIC ALGOL ALGORITHM ALPHA ALPHABETIC
ALPHANUMERIC ALTERNATIVE AMPLITUDE ANALOGUE ANALYSIS ANGLE
     MORE?
*Yes.
     ANNUITY ANOVA ANSWER APPROXIMATE APPROXIMATION AREA ARGUMENT
ARITHMETIC ARRAY ARRIVAL ASCENDING ASSAY ASSEMBLER ASSOCIATION
ASSUMPTION ASYMMETRICAL ASSYMPTOTICALLY-EFFICIENT ASYMPTOTICAL
ASYMPTOTIC AUGMENTED-MATRIX
     MORE?
*Yes.
     AUTOCORRELATION AUTOCOVARIANCE AUTO-SPECTRA AVERAGE AXIS BAL
BALANCED BALANCE-SHEET BANK BARTLETT BARTOS BAR-PLOT BASE-E BASE-10
BASE-2 BASHARIN BASIC BATCH BAUMANN BAYESIAN
     MORE?
*No.
*How many analysis programs are there?
     106
*How many of them handle matrices?
     34
*Which of those are written in Fortran?
     CAP CLUSTER CORREL DATSIM FACTAN FSCORE IUFAC LAG MDSCAL MULCVR
SUBMTX TSSA
*What are cluster's properties?
     ADDITIVE-MODEL ANALYSIS CLUSTERING COEFFICIENT CORRELATION ETA
FACTORS FORTRAN GROUP GUTTMAN IBM360/40 IBM360/67 MATRICES
MULTIDIMENSIONAL PHI PHIMAX RESULT SCALE SPACE
*Are there any other clustering programs written in Fortran?
     YES.
*What computers do they run on?
     CAP : IBM360/40 IBM360/67
     MULTYP : CDC6400 DEC-PDP10 UNIVAC1108
     OSIRIS-II-LEVEL-2 : IBM360/40 IBM360/67
*Is Cobol a language?
     NO.
*What languages are available?
     ALGOL ASSEMBLER BAL BASIC PL/1 SNOBOL4 SPITBOL FORTRAN
FORTRAN-II FORTRAN-IV FORTRAN-V FORTRAN-66 LISP WATFIV WATFOR WATIV
*What computers support Algol programs?
     BURROUGHS5500
*Does the Burroughs5500 accept programs not written in Algol?
     ONE MOMENT PLEASE.
     NO.
*Give me some programs written in Pl/1 as well as Fortran-iv.
     THERE ARE NONE.
*Are there any Bal programs?
     YES.
*What are they?
     COPY EDIT ERROR ITEM MATCH MERGE MERMAC QUEST RECODE SCORE
SELECT SEQUENCE SORT TOTAL
*How many properties are associated with each one?
     COPY : 6
     EDIT : 7
     ERROR : 10
     ITEM : 21
     MATCH : 8
     CONTINUE?
*No.
*List all of the properties of the Bal programs.
     COPY : BAL DISK DRUM FILE IBM360 TAPE
     EDIT : BAL DISK DRUM FIELD IBM360 RECORD TAPE
     ERROR : BAL DATA ERRORS IBM360 ITEMS LISTING MULTIPLE-CHOICE
```

SCORING STUDENT TEST
ITEM : ACHIEVED ALTERNATIVE BAL BISERIAL CORRELATION DATA DISTRIBUTION FREQUENCY HISTOGRAM IBM360 INDIVIDUAL ITEMS PERCENTAGE POINT PROPORTION RAW RESPONSE SCORING STANDARD STATISTIC TEST
MATCH : BAL COMPARISON FIELD FILE GROUP IBM360 MATCHING RECORD CONTINUE?
*Yes.
MERGE : ASCENDING BAL DISK DRUM FIELD FILE GROUP IBM360 ORDER RANDOM RECORD TAPE
MERMAC : BAL DATA IBM360 MANIPULATION TEST
QUEST : ANSWER BAL DATA DECIMAL DEVIATION DISTRIBUTION FRACTION FREQUENCY IBM360 ITEMS MEAN NEGATIVE NUMBER STANDARD TEST VALUE WEIGHT WEIGHTED
RECODE : BAL CHARACTER DISK DRUM IBM360 RECORD SCHEME TAPE TRANSFORM
SCORE : BAL DATA DECIMAL FRACTION IBM360 ITEMS NEGATIVE NUMBER RESPONSE SAMPLE SCORING SUBJECT SUBSCORE VALUE WEIGHT WEIGHTED CONTINUE?
*Yes.
SELECT : ALTERNATIVE ANSWER BAL BISERIAL BROWN CORRELATION CRITERION DATA DEVIATION DISTRIBUTION EXTERNAL FREQUENCY HISTOGRAM IBM360 ITEMS KR-21-RELIABILITY MEAN MEASURE MULTIPLE-CHOICE POINT PROPHESY PROPORTION SCORING SPEARMAN STANDARD TEST
SEQUENCE : BAL CHARACTER DISK DRUM ERRORS FIELD IBM360 IDENTIFICATION MATCHING PROPER RECORD TAPE
SORT : ASCENDING BAL DISK DRUM FIELD IBM360 RECORD TAPE
TOTAL : BAL BROWN DECIMAL DEVIATION DISTRIBUTION FRACTION FREQUENCY HISTOGRAM IBM360 KR-21-RELIABILITY MEAN NEGATIVE NUMBER PROPHESY RANGE SCORING SPEARMAN STANDARD TEST
*Please list all of the computers for me.
BURROUGHS5500 CDC1604 CDC3400 CDC3600 CDC3600-COMPASS CDC6000 CDC6400 CDC6500 CDC6600 DEC-PDP10 GE635 IBM1130 IBM1401 IBM1620 IBM1802 IBM360 IBM360/40 IBM360/50 IBM360/65 IBM360/67 IBM360/91 IBM370/145 IBM7090 IBM7040 IBM709 IBM7094 RCA-SPECTRA70 RCA/2-7 SPECTRA70 UNIVAC-SDF UNIVAC1105 UNIVAC1108-ASSEMBLER UNIVAC1108-EXECS
*

APPENDIX II

KEYWORD LISTING

Please list all of the properties.

ACCESS ACHIEVED ADDITIVE-MODEL ADJACENT ADJUSTED AGE AGGREGATED
AGREEMENT AIKEN ALGEBRAIC ALGOL ALGORITHM ALPHA ALPHABETIC
ALPHANUMERIC ALTERNATIVE AMPLITUDE ANALOGUE ANALYSIS ANGLE ANNUITY
ANOVA ANSWER APPROXIMATE APPROXIMATION AREA ARGUMENT ARITHMETIC ARRAY
ARRIVAL ASCENDING ASSAY ASSEMBLER ASSOCIATION ASSUMPTION ASYMMETRICAL
ASSYMPTOTICALLY-EFFICIENT ASYMPTOTICAL ASYMPTOTIC AUGMENTED-MATRIX
AUTOCORRELATION AUTOCOVARIANCE AUTO-SPECTRA AVERAGE AXIS BAL BALANCED
BALANCE-SHEET BANK BARTLETT BARTOS BAR-PLOT BASE-E BASE-10 BASE-2
BASHARIN BASIC BATCH BAUMANN BAYESIAN BCD BCDIC BEHAVIOR
BESSEL-FUNCTION BEST-FIT BETA BETA1 BETA2 BHAPKAR BIASED BINARY
BINOMIAL BIOASSAY BIOLOGICAL BIOMEDICAL BIOSTATISTICAL BIQUARTIMIN
BIRTH BISERIAL BIVARIATE BLANK BLOCK BLOCKED BLOCKSIZE BOAS BOOKLET
BOOLEAN-EXPRESSION BOOLEAN-SELECTION-OF-CASES BORGATTA BOUND
BRACKETING BRANCHING BRILLOUIN BROWN BURROUGHS5500 BUSINESS CAI
CALCOMP CALCULATION CALLING CANONICAL CAPABILITY CAPACITY CARD CASE
CASEWISE CASE-COMBINATIONS CASE-COUNT CASE-DECILES CASE-NUMBER
CASH-BUDGET CATEGORICAL CAUSAL CDC CDC1604 CDC3400 CDC3600
CDC3600-COMPASS CDC6000 CDC6400 CDC6500 CDC6600 CELL CELL-COUNT
CELL-SIZE CENSORED CENTERED CENTERING CENTRAL-LIMIT-THEOREM CHANGE
CHARACTER CHARACTERISTIC CHECKER CHECKING CHILTON CHI-SQUARE
CIRCULAR-DISTRIBUTION CLASSIFICATION CLASSIFIED CLASSROOM
CLASSROOM-LEARNING-SITUATION CLASS-INTERVAL C-LEVEL CLIMATE CLOSED
CLUSTERING COBWEB COCHRANE-ORCUTT COCHRAN CODE CODE-BOOK CODE-CHECK
CODING COEFFICIENT COHERENCE COLUMN COMBINATION COMMODITY COMMON
COMMUNALITY COMPARISON COMPASS COMPLETE COMPLEX COMPONENT COMPOSITION
COMPUTER-PROBLEM-BOOKLET CONCEPTUAL CONCORDANCE CONDITIONED
CONFIDENCE CONFIGURATION CONFUSION CONSENSUS CONSISTENCY CONSTRAINT
CONSTRUCTION CONSUMPTION CONTINGENCY CONTINUITY CONTINUOUS CONTRAST
CONTRIVED CONTROL CONVERSATIONAL CONVERSION COOMBS-NONMETRIC-SCALING
COORDINATE COPYING CORNELL CORRELATION COSINE COST COUNTS COVARIANCE
COVARIATE COVARIMIN CRAMER-V CRAP-GAME CRITERION CRITICAL CROSS
CROSSED CUBIC CUMULATIVE CURVE CUT-POINT CYCLE CYCLICAL DATA DATASET
DATA-ARRANGING DATA-CASE DATE DEAD DEC DEC-PDP10 DECILE DECIMAL
DECISION DECK DEDUCTIVE DEFINITE DEGREE DEGREE-M DELETION DELIVERY
DEMAND DEMAND-REGION DENOMINATOR DENSITY DEPENDENT DERIVATIVE
DESCENDING DESCRIPTION DESCRIPTIVE DESIGNATED DESIGN DETECTOR
DETERMINANT DETERMINATION DEVIATION DEVICE DIAGNOSIS DIAGONAL DIAGRAM
DICHOTOMY DICTIONARY DIE DIFFERENCE DIFFERENCING DIFFERENTIABLE
DIFFERENTIAL DIMENSION DIMENSIONED DIMINISHING DISCRETE
DISCRETE-PRIOR DISCRIMINANT DISK DISPATCHING DISPERSION DISSIMILARITY
DISTANCE DISTRIBUTED DISTRIBUTION DIVERSITY DOUBLE DOUBLE-PRECISION
DRAG DRAW DRILL DRUM DUMMY DUNCAN DUPLICATE DURBIN DYNAMIC EBCDIC
ECONOMETRIC ECONOMIC ECONOMY EDITING EDUCATION EFFECT EIGENVALUE
EIGENVECTOR ELASTICITY ELEMENT ELEMENTARY ELIMINATION EMPIRICAL
ENDOGENOUS ENDPOINT END ENGLISH ENP ENTITY EQUATION EQUIDISTANT
EQUIMAX ERRORS ERROR-OF-ESTIMATE ERROR-OF-MEAN ESTIMATE ESTIMATED
ESTIMATES-OF-ERROR ESTIMATION ESTIMATOR ETA EVENNESS EXACT EXCLUSION
EXCLUSIVE EXECUTABLE EXECUTE EXOGENOUS EXPECTATION EXPECTED
EXPERIMENTAL EXPERIMENT EXPONENTIAL EXTERNAL EXTRANEOUS FACTORS
FACTORIAL FACTORING FACTOR-BY-FACTOR FAIL FIELD FILE FILE-CORRECTION
FILTER FILTERED FINAL FINANCIAL FINANSIM FINITE FIRM FIRST-DERIVATIVE
FIRST-ORDER FIRST-TRIGONOMETRIC-MOMENT FISCAL FISHER FIT FIXED
FLOATING FORCED FORD FORECAST FORM FORMAT FORMULATED FORMULATION
FORTRAN FORTRAN-II FORTRAN-IV FORTRAN-V FORTRAN-63 FOURIER
FOUR-DIMENSIONAL FRACTILE FRACTION FRANCIS FREEDOM FREQUENCY FRIEDMAN
FTAU99 FULKERSON FULL FUNCTION FUTURE F-CURVE F-DENSITY F-DISTRIBUTED
F-LEVEL F-RATIO F-STATISTIC F-TABLE F-TEST F-TO-ENTER F-TO-REMOVE
F-VALUE GAME GAMMA GAUSS GAUSSIAN GE GENERALIZED GENERAL-PURPOSE
GENERATION GENERATOR GEOGRAPHY GEOMETRIC GE635 GINI-COEFFICIENT
GOLDFARB-VARIABLE-METRIC-EXTENSION GOMPERTZ GOODMAN-KRUSKAL-GAMMA
GOODNESS-OF-FIT GOVERNMENT-SPENDING GRADIENT GRAND GRAPHS GROUP

HARRINGTON-HASTINGS-APPROXIMATION HEADED HEADING HEISENBERG
HICKS-HANSEN HIERARCHICAL HISTOGRAM HOMOGENEITY
HOMOGENEITY-OF-VARIANCE-TEST HOMOGENEOUS HORIZONTAL HOTELLING
HOUSEHOLDER HYPERGEOMETRICAL HYPOTHESIS IBM IBM1130 IBM1401 IBM1620
IBM1802 IBM360 IBM360/40 IBM360/50 IBM360/65 IBM360/67 IBM360/91
IBM370/145 IBM7090 IBM7040 IBM709 IBM7094 IDENTICAL IDENTIFICATION
ILLEGAL IMAGE IMPACT IMPORT IMPROVEMENT INCOME INCOMPLETE INCORRECT
INCREASE INCREASING INDEPENDENCE INDEPENDENT INDEX INDIVIDUAL
INDUCTIVE INEQUALITY INFINITY INFORMATION-CONTENT INFORMATION-MATRIX
INFORMATION-RETRIEVAL INFORMATION-THEORETICAL INITIAL INPUT INSERTION
INSIGNIFICANT INTEGER INTEGRAL INTEGRATED INTERACTION INTERACTIVE
INTERCEPT INTERDECILE INTERDEPENDENT INTERDISCIPLINARY INTEREST
INTERFACE INTERITEM INTERMEDIATE INTERNAL INTERPAIR INTERPOINT
INTERPRETATION INTERQUARTILE INTERVAL INTRACLASS INTRINSIC INVALID
INVERSE INVERSION INVESTMENT ITEMS ITERATION ITERATIVE ITH
I/O-ROUTINE JENKINS JOHNSON JOHNSTON JOINT JORDON KENDALL KEYNESIAN
KIEFER KOLMOGOROV KOPPEN KRUSKAL-WALLIS KRUSKAL-WALLIS-H
KR-21-RELIABILITY KURTOSIS K-CLASS K-SAMPLE LABEL LABELED LABELING
LAGS LAPLACE LATIN-SQUARE LEAD LEAKAGE LEAL LEAST-SQUARE LEGISLATIVE
LEGITIMATE LENGTH LEVEL LIBRARY LIFE LIKELIHOOD-FUNCTION LIMITED
LIMITED-INFORMATION LINE LINEAR LINEARITY LINE-PRINTER LISP LISTS
LISTING LM-CURVES LOADING LOBE LOENINGER LOG LOGARITHM LOGIC LOCICAL
LOGISTIC LORENZ LOSS LUMSDEN MACHINE-READABLE MACRO MACROECONOMIC
MAGNETIC MAHALANOBIS MAINLINE MANAGEMENT-SIMULATION MANIPULATION
MANNER MANN-WHITNEY MAP MARGINAL MARKER MATCHED MATCHING
MATCH-MERGING MATHEMATICAL MATRICES MAXIMIZING MAXIMUM
MAXIMUM-LIKELIHOOD MCMEAN MCNEMAR MCR MEAN MEASURE MEASUREMENT
MECHANICAL MEDIAN MEDITERRANEAN MEIER MEMORY-SYSTEM MENZEL MERGES
METHOD METRIC MIDTEX MILEAGE MINIMIZATION MINIMUM MISSING MODE MODEL
MODIFIED MOMENT MONCREIFF MONTE-CARLO MOORE MOSES MOVING-AVERAGE
MUCHMORE MULTIDIMENSIONAL MULTINOMINIAL MULTIPASS MULTIPLE
MULTIPLE-CHOICE MULTIVARIATE MUTUAL NATURAL NEGATIVE NESTED NETWORK
NEWTON NOISE NOMINAL NONCENTRALITY NONPARAMETRIC NONRANDOM
NONSINGULAR NONSYMMETRICAL NON-LINEAR NON-MISSING NON-NEGATIVE
NON-OSIRIS NON-RANDOMNESS NON-SIGNED NON-STANDARD NORMAL NORMALIZED
NORMING NULL-HYPOTHESIS NUMBER NUMBERED NUMERATOR NUMERIC NWAY
N-DIMENSIONAL OBJECTIVE OBJECT OBLIMIN OBLIQUE OBSERVATION OCCURRENCE
OCDQ ODD OLS OLSADD OLSDEL OLSHC ONEWAY ONE-COLUMN ONE-PAGE
ONE-SAMPLE ONE-SIDED ONE-TAILED ONION OPEN-ENDED OPINION OPTIMAL
ORDER ORDINAL ORDINARY ORGANIZATION ORIGIN ORTHANT ORTHOGONAL OSIRIS
OS/360 OUTLIER OUTPUT OVERLAY PACKAGE PAGE PAIR PAIRED PANEL PAPER
PARABOLIC PARALLEL PARAMETER PARAMETRIC PARTIAL PARTITION PART-WHOLE
PASCAL PASS PASSING PATTERN PDP PDP10 PEARSON PEARSONIAN PENALTY
PERCENTAGE PERFORMANCE PERIOD PERIODIC PERT PHASE PHI PHIMAX PICTURE
PIVOTAL PLACEMENT PLACE PLANNING PLOT PL/1 POINT POISSON POLICY
POLITICS POLYCHOTOMY POLYNOMIAL POOLED POPULATION PORTION POSITION
A-POSTERIORI POTENTIAL POWER PRACTICE PRECISE PRECISION PREDETERMINED
PREDICTED PREDICTOR PREPARATION PREPROCESSOR PRESCRIBED PRICE
PRINCIPAL PRINTED PRINTER PRINTOUT PRINT-PLOT PROBABILISTIC PROBITS
PROBLEM PROBLEM-PARAMETERS PROCEDURE PROCESS PRODUCT PRODUCTION
PROFILES PROFIT-MAXIMIZATION PROGRESSIVE PROJECTED PROJECTION PROPER
PROPHESY PROPORTION PROXIMITY PSEUDO PSYCHOLOGY PULSE PUNCH QUADRATIC
QUALITATIVE QUANTITATIVE QUANTITY QUARTIMAX QUARTIMIN QUESTIONNAIRE
QUESTION QUEUE QUICK Q-TEST RADIAN RAINFOREST RAJU RANDOM RANGE RANK
RAO RAPHSON RATE RATIO RATIONAL RAW RAYLEIGH RCA RCA-SPECTRA70
RCA/2-7 REACTION READING REAL REASONING RECODING RECORD RECTANGULAR
RECURSIVE REDUCED REDUNDANCY REFLECTION REFORMAT REGION REGRESSIONS
RELATED RELATIONSHIP RELATIVE RELAXATION RELIABILITY REPEATED REPLICA
REPORT REPRESENTATION REPRODUCIBILITY REPRODUCTION RESEARCH RESIDUAL
RESIDUE RESOLUTION RESPONSE RESTRICTION RESULT REVENUE REVERSE REVIEW
REWINDING RHO RIGHT RIGHT-HAND-SIDE RISK ROLL-CALL ROOTS ROSEN
ROTATED ROUND-OFF ROUTE ROW RULE SAMPLE SAMPLING SATTERTHWAITE
SCALABILITY SCALAR SCALE SCALING SCALOGRAM SCANNING SCATTER SCHEFFE
SCHEME SCHLAIFER SCHUESSLER SCIENCE SCIENTIFIC SCORING SCREENING
SEASONAL SECOND-DERIVATIVE SECOND-MOMENTS SECOND-ORDER
SECOND-TRIGONOMETRIC-MOMENT SECTION SEGMENTED SEIDEL SEMANTIC

SEPARATION SEQUENCES SEQUENTIAL SERIAL SERIES SERVICE SET SETUP SEX
SHANNON SHIRER SIGN SIGNED SIGNIFICANCE SIGNIFICANT SIMPLE SIMPSON
SIMULATED SIMULTANEOUS SINE SINGLE SINGULARITY SIR SIZE SKEWNESS
SMIRNOV SNEDECOR SNOBOL4 SOCIAL SOCRATIC SOIL SOLUTION SOLVER SOMER-D
SOMER-DYX-&-DXY SORTS SORTER SPACE SPATIAL SPEARMAN SPECIAL SPECIES
SPECIFIED SPECTRAL SPECTRA70 SPECTRUM SPITBOL SPLIT-SPLIT SPS SQUARE
SQUARED STABILIZATION STACKED STANDARD STANDARDIZED STAND-ALONE
STARTING STATE STATEMENT STATION STATIONARY STATISTIC STATUS STEPWISE
STRAIGHT STRATA STRING STRUCTURAL STRUCTURE STUART STUDENT STUDY
STURGE SUBGROUP SUBJECT SUBPOPULATION SUBSAMPLE SUBSCORE SUBSET
SUBSTITUTION SUBTEST SUBMATRIX SUCCESS SUM SUMMARY SUMMATION SUPPLY
SURVEY SURVIVAL SYMBOL SYMMETRIC TABLE TABLE-LOOKUP TABLE-WIDE
TABULAR TABULATION TAIL TAPE TAU TEACHING TECHNICAL TECHNIQUE TERM
TERMINAL TEST TETRACHORIC THEORETICAL THEORY THREEWAY THREE-COLUMN
THREE-DIMENSIONAL THREE-FACTOR THREE-STAGE THRESHOLD TIED TIME
TIME-LOCKED TIME-OF-DAY TOBIN TOLERANCE TRANSFER-FUNCTION
TRANSFER-VARIABLE TRANSFORM TRANSGENERATED TRANSITION TRANSLATED
TRANSPORTATION TRANSPOSED TRAU TREATMENT TREE TRIAL TRIANGLE
TRICHOTOMOUS TRIPLE TRUCK TUKEY-FILTER TUTORIAL TWOWAY TWO-COLUMN
TWO-DIMENSIONAL TWO-DIMENSIONED TWO-SAMPLE TWO-SIDED TWO-STAGE
TWO-TAIL TYPAL TYPE TYPEWRITER TYPE-I TYPE-II T-DENSITY T-DISTRIBUTED
T-PROGRAM T-SQUARE T-STATISTICS T-TEST T-VALUE UNBALANCED UNBIASED
UNCERTAINTY UNCLASSIFIED UNCONSTRAINED UNCORRELATED UNDERFLOW
UNDERLYING UNFORMATTED UNGROUPED UNIDIMENSIONAL UNIFORM UNIT UNITY
UNIVAC UNIVAC-SDF UNIVAC1108 UNIVAC1108 UNIVAC1108-EXEC8 UNIVARIATE
UNKNOWN UNMATCHED UNWEIGHTED UPDATING UPPER UTILITY UWHAUS
U-STATISTIC U-TEST VALID VALUE VARIABLE VARIANCE VARIATE VARIATION
VARIMAX VECTOR VEGETATION VERSION VERTICAL VOLUME-LABEL WALD WALLACE
WALSH WATFIV WATFOR WATIV WATSON WATTS WAVE WEIGHT WEIGHTED WICKLUND
WIDE-BAND WIELANDT WILCOXON WILD WILKS-LAMBDA WILLIAMS WILSON
WOLFOWITZ WORLD XLWR XUPR X-ARRAY X-AXIS X-SERIES X-VALUE X-VARIABLE
YATES YNORM YULE Y-AXIS Y-INTERCEPT Y-VALUE ZELLNER ZERO ZERO-ORDER
Z-SCORE Z-TRANSFORMATION Z-VALUE
*

APPENDIX III

PROGRAM LISTING

```
(ENDA LAP                                                          00000100
 (((FUNCTION CLOSER2A (W1 W2)                                      00000200
     (C AC (ENTRY CHE))                                            00000300
     (BC LQ (LABEL W2CH))                                          00000400
     (LH AC (4 AC SORG))                                           00000500
     (NR AC MASK)                                                  00000600
     (SLL AC (2))                                                  00000700
     (LH AC3 (0 AC SORG))                                          00000800
     (N AC3 (NUMBER 1FFFX))                                        00000900
     (LA AC (4 AC SORG))                                           00001000
     (GO L1)                                                       00001100
     W2CH (LA AC3 (1))                                             00001200
     (S AC (ENTRY CHO))                                            00001300
     (SRL AC (2))                                                  00001400
     (STC AC (ENTRY TEMP 4))                                       00001500
     (LA AC (ENTRY TEMP 4))                                        00001600
     L1 (L AC0 W1)                                                 00001700
     (C AC0 (ENTRY CHE))                                           00001800
     (BC LQ (LABEL W1CH))                                          00001900
     (LH AC0 (4 AC0 SORG))                                         00002000
     (NR AC0 MASK)                                                 00002100
     (SLL AC0 (2))                                                 00002200
     (LH AC2 (0 AC0 SORG))                                         00002300
     (N AC2 (NUMBER 1FFFX))                                        00002400
     (LA AC0 (4 AC0 SORG))                                         00002500
     (GO L2)                                                       00002600
     W1CH (LA AC2 (1))                                             00002700
     (S AC0 (ENTRY CHO))                                           00002800
     (SRL AC0 (2))                                                 00002900
     (STC AC0 (ENTRY TEMP))                                        00003000
     (LA AC0 (ENTRY TEMP))                                         00003100
     L2 (CR AC2 AC3)                                               00003200
     (BC LQ (LABEL L3))                                            00003300
     (LR AC2 AC3)                                                  00003400
     L3 (BCTR AC2 0)                                               00003500
     (EX AC2 (LABEL CLC))                                          00003600
     (BC L (LABEL W1))                                             00003700
     (BC G (LABEL W2))                                             00003800
     (BCTR AC3 0)                                                  00003900
     (CR AC2 AC3)                                                  00004000
     (BC GQ (LABEL W2))                                            00004100
     W1 (L AC W1)                                                  00004200
     (GO OUT)                                                      00004300
     CLC (CLC 0 (0 AC0) (0 AC))                                    00004400
     W2 (L AC W2)                                                  00004500
     OUT)))                                                        00004600
  DEFINE                                                           00004700
  (((SORT (LAMBDA (L)                                              00004800
      (PROG (HERE TEMP FLAG)                                       00004900
       (COND ((LESSP (LENGTH L) 2) (RETURN L)))                    00005000
       (SETQ FLAG 1)                                               00005100
       (SETQ L (CONS NIL L))                                       00005200
       A (COND ((EQ FLAG NIL) (RETURN (CDR L))))                   00005300
       (SETQ HERE L)                                               00005400
       (SETQ FLAG NIL)                                             00005500
       B (SETQ TEMP (CADR HERE))                                   00005600
       (COND ((EQ TEMP (CLOSER2A TEMP (CADDR HERE))) (GO C)))      00005700
       (SETQ TEMP (CDR HERE))                                      00005800
       (RPLACD HERE (CDR TEMP))                                    00005900
       (RPLACD TEMP (CDDR HERE))                                   00006000
```

```
        (RPLACD (CDR HERE) TEMP)                                        00006100
        (SETQ FLAG 1)                                                   00006200
        C (SETQ HERE (CDR HERE))                                        00006300
        (COND ((NOT (NULL (CDDR HERE))) (GO B)))                        00006400
        (GO A))))))                                                     00006500
    DEFINE                                                              00006600
    (((MAKEATOM (LAMBDA (L)                                             00006700
        (BLOCK ((RLINE . 124)                                           00006800
         (NEXTCH . 124)                                                 00006900
         ((CURCOLR . 124) 1)                                            00007000
         ((LASTCOLR . 124) (ADDSMALL (LENGTH L) 1)))                    00007100
        (SETQ (RLINE . 124)                                             00007200
         ((GETARRAY . 122) (GEXP STRING) (LASTCOLR . 124) NIL))         00007300
        (FOR I EQ 1 C IN L DO ((SETC . 122) (RLINE . 124) I C))         00007400
        ((SETC . 122) (RLINE . 124) (LASTCOLR . 124) *SPACE*)           00007500
        (RETURN (RATOM))))))))                                          00007600
    CSET (LETTERS (A B C D E F G H I J K L M N O P Q R S T U V W X Y Z)) 00007700
    EVAL ((CSETQ LETTERS (CONS *MINUS* (CONS *SLASH* LETTERS))))        00007800
    CSET (DIGITS ($$$0$ $$$1$ $$$2$ $$$3$ $$$4$ $$$5$ $$$6$ $$$7$ $$$8$ 00007900
      $$$9$))                                                           00008000
    CSET (*QMARK* ?)                                                    00008100
    CSET (*EXMARK* .)                                                   00008200
    CSET (*3DOTS* ...)                                                  00008300
    EVAL ((CSETQ ENDS (LIST *QMARK* *EXMARK* *DOT*)))                   00008400
    DEFINE                                                              00008500
    (((EOLP (LAMBDA NIL (GREATERP (CURCOLR . 124) (LASTCOLR . 124))))))  00008600
    DEFINE                                                              00008700
    (((ENDSENTENCE (LAMBDA NIL (PROG (C)                                00008800
        (COND ((EOLP) (RETURN NIL))                                     00008900
         ((EQ (SETQ C (READCH)) *SPACE*) (GO A))                        00009000
         (T (RETURN (LIST C))))                                         00009100
        A (COND ((EOLP) (RETURN NIL)))                                  00009200
        (COND ((EQ (SETQ C (READCH)) *SPACE*) (GO A))                   00009300
         (T (RETURN (LIST C *SPACE*))))))))))                           00009400
    DEFINE                                                              00009500
    (((CONREAD (LAMBDA NIL (PROG (ANS TEMP)                             00009600
        CLEAR (TERFAD)                                                  00009700
        ZIP (COND ((EQ (CAR (SETQ ANS (LIST (READCH)))) *SPACE*)        00009800
          (GO ZIP)))                                                    00009900
        (GO B)                                                          00010000
        A (COND ((EOLP) (SETQ ANS (CONS *SPACE* ANS))))                 00010100
        C (SETQ ANS (CONS (READCH) ANS))                                00010200
        B (COND ((MEMBER (CAR ANS) ENDS)                                00010300
          (COND                                                         00010400
           ((OR (NULL (SETQ TEMP (ENDSENTENCE)))                        00010500
             (NOT (EQ (CAR ANS) *DOT*)))                                00010600
            (RETURN (REVERSE ANS)))                                     00010700
           (T (PROGN (SETQ ANS (APPEND TEMP ANS)) (GO B)))))            00010800
         ((AND (EQ (CAR ANS) *MINUS*) (EOLP))                           00010900
          (PROGN (SETQ ANS (CDR ANS)) (GO C)))                          00011000
         ((AND (EQ (CAR ANS) *SPACE*) (EQ (CADR ANS) *SPACE*))          00011100
          (SETQ ANS (CDR ANS)))                                         00011200
         ((EQ (CAR ANS) (QUOTE :)) (RPLACA ANS (READ))))                00011300
        (GO A))))))                                                     00011400
    DEFINE                                                              00011500
    (((SCANNER (LAMBDA (S)                                              00011600
        (PROG (ANS TEMP)                                                00011700
        A (COND ((NULL S) (RETURN (REVERSE ANS))))                      00011800
        (SETQ TEMP S)                                                   00011900
        B (COND ((MEMBER (CAR S) LETTERS)                               00012000
```

```
(SETQ ANS (CONS (PROG (HERE)                                              00012100
    L1 (COND                                                              00012200
     ((OR (MEMBER (CADR S) LETTERS)                                       00012300
          (MEMBER (CADR S) DIGITS)                                        00012400
          (AND (EQ (CADR S) *DOT*) (CDDR S)))                             00012500
      (PROGN (SETQ S (CDR S)) (GO L1))))                                  00012600
     (SETQ HERE S)                                                        00012700
     (SETQ S (CDR S))                                                     00012800
     (RPLACD HERE NIL)                                                    00012900
     (RETURN (COMPRESS TEMP)))                                            00013000
    ANS)))                                                                00013100
((MEMBER (CAR S) DIGITS)                                                  00013200
 (SELECT (PROG (HERE)                                                     00013300
   N1 (COND ((MEMBER (CADR S) DIGITS)                                     00013400
     (PROGN (SETQ S (CDR S)) (GO N1)))                                    00013500
     ((MEMBER (CADR S) LETTERS)                                           00013600
      (PROGN (SETQ S (CDR S)) (RETURN 2)))                                00013700
     ((EQ (CADR S) *DOT*)                                                 00013800
      (COND ((NULL (CDDR S))                                              00013900
        (PROGN (SETQ HERE S)                                              00014000
         (SETQ S NIL)                                                     00014100
         (RPLACD HERE NIL)                                                00014200
         (SETQ ANS (CONS *DOT* (CONS (MAKEATOM TEMP) ANS)))               00014300
         (RETURN 0)))                                                     00014400
       ((MEMBER (CADR (SETQ S (CDR S))) DIGITS)                           00014500
        (PROGN (SETQ ANS (CONS (PROG (HERE)                               00014600
            N2 (COND ((MEMBER (CADR (SETQ S (CDR S))) DIGITS)             00014700
              (GO N2)))                                                   00014800
           (SETQ HERE S)                                                  00014900
           (SETQ S (CDR S))                                               00015000
           (RPLACD HERE NIL)                                              00015100
           (RETURN (MAKEATOM TEMP)))                                      00015200
          ANS))                                                           00015300
        (RETURN 0)))                                                      00015400
       (T (PROGN (SETQ HERE S)                                            00015500
         (SETQ S (CDR S))                                                 00015600
         (RPLACD HERE NIL)                                                00015700
         (SETQ ANS (CONS (MAKEATOM TEMP) ANS))                            00015800
         (RETURN 0)))))                                                   00015900
     (T (PROGN (SETQ HERE S)                                              00016000
       (SETQ S (CDR S))                                                   00016100
       (RPLACD HERE NIL)                                                  00016200
       (SETQ ANS (CONS (MAKEATOM TEMP) ANS))                              00016300
       (RETURN 0)))))                                                     00016400
  (2 (GO B))                                                              00016500
  NIL))                                                                   00016600
((EQ (CAR S) *SPACE*) (SETQ S (CDR S)))                                   00016700
((EQ (CAR S) *DOT*)                                                       00016800
 (COND ((NULL (CDR S))                                                    00016900
   (PROGN (SETQ ANS (CONS *DOT* ANS)) (SETQ S NIL)))                      00017000
   ((MEMBER (CADR S) DIGITS)                                              00017100
    (SETQ ANS (CONS (PROG (HERE)                                          00017200
       D1 (COND ((MEMBER (CADR S) DIGITS)                                 00017300
         (PROGN (SETQ S (CDR S)) (GO D1))))                               00017400
       (SETQ HERE S)                                                      00017500
       (SETQ S (CDR S))                                                   00017600
       (RPLACD HERE NIL)                                                  00017700
       (RETURN (MAKEATOM (CONS (CAR DIGITS) TEMP))))                      00017800
      ANS)))                                                              00017900
  ((EQ (CADR S) *DOT*)                                                    00018000
```

```
            (COND ((NULL (CDDR S))                                          00018100
              (PROGN (SETQ ANS (CONS *DOT* ANS)) (SETQ S NIL)))             00018200
             ((EQ (CADDR S) *DOT*)                                          00018300
              (PROGN (SETQ ANS (CONS *3DOTS* ANS)) (SETQ S (CDDDR S))))     00018400
             (T (PROGN (SETQ ANS (CONS *DOT* ANS)) (SETQ S (CDR S))))))     00018500
           (T (PROGN (SETQ ANS (CONS *DOT* ANS)) (SETQ S (CDR S))))))       00018600
         (T (PROGN (SETQ ANS (CONS (CAR S) ANS)) (SETQ S (CDR S)))))        00018700
        (GO A))))))                                                         00018800
  DEFINE                                                                    00018900
  (((NCDR (LAMBDA (L N)                                                     00019000
      (PROG NIL A (COND ((LESSP N 1) (RETURN L)))                           00019100
        (SETQ L (CDR L))                                                    00019200
        (SETQ N (SUB1 N))                                                   00019300
        (GO A))))))                                                         00019400
  DEFINE                                                                    00019500
  (((UNION1 (LAMBDA (L1 L2)                                                 00019600
      (PROG NIL (SETQ L1 (NCONC L1 L2))                                     00019700
        (SETQ L2 NIL)                                                       00019800
        A (COND ((NULL L1) (RETURN (DREVERSE L2)))                          00019900
         ((MEMB (CAR L1) (CDR L1)) NIL)                                     00020000
         (T (SETQ L2 (CONS (CAR L1) L2))))                                  00020100
        (SETQ L1 (CDR L1))                                                  00020200
        (GO A))))))                                                         00020300
  DEFINE                                                                    00020400
  (((COPY1 (LAMBDA (L)                                                      00020500
      (PROG (CL)                                                            00020600
        A (COND ((NULL L) (RETURN (DREVERSE CL))))                          00020700
        (SETQ CL (CONS (CAR L) CL))                                         00020800
        (SETQ L (CDR L))                                                    00020900
        (GO A))))))                                                         00021000
  DEFINE                                                                    00021100
  (((INTERSECTION1 (LAMBDA (L1 L2)                                          00021200
      (PROG (L3)                                                            00021300
        A (COND ((NULL L1) (RETURN (DREVERSE L3)))                          00021400
         ((AND (MEMB (CAR L1) L2) (NOT (MEMB (CAR L1) L3)))                00021500
          (SETQ L3 (CONS (CAR L1) L3))))                                    00021600
        (SETQ L1 (CDR L1))                                                  00021700
        (GO A))))))                                                         00021800
  DEFINE                                                                    00021900
  (((UNLIST (LAMBDA (E)                                                     00022000
      (COND ((ATOM E) (LIST E))                                             00022100
        (T (PROG ((K (LIST NIL)))                                           00022200
          BACK (COND ((ATOM E) (RETURN (CAR K))))                           00022300
           (SETQ K (LCONC (UNLIST (CAR E)) K))                              00022400
           (SETQ E (CDR E))                                                 00022500
           (GO BACK)))))))))                                                00022600
 (ENQB SPECIAL                                                              00022700
  ((SENT PUTS OBJ NEG ATT GRP OFF TEMP ABSFILE RESULT ERRORS))              00022800
  CSET (STOPPERS (FAST DOCUMENTED SYSTEM SYSTEMS REDISTRIBUTABLE            00022900
    MANAGEMENT MODULARIZED WHERE TOPIC TOPICS DOCUMENTATION SOURCE          00023000
    DISTRIBUTOR DISTRIBUTORS DISTRIBUTES EXAMPLE EXAMPLES CORE BIG          00023100
    SUBROUTINE SUBROUTINES MORE GREATER LESS))                              00023200
  CSET (BLOCKERS (ONLY THIS PERCENT $$$+$ # $ * : . @ ; .))                 00023300
  EVAL                                                                      00023400
  ((CSETQ BLOCKERS (CONS *PERCENT* (CONS *LBRAC* (CONS *RBRAC* 'BLOCKERS)))00023500
)))                                                                         00023600
  CSET (ON T)                                                               00023700
  CSET (CONTINUE 5)                                                         00023800
  CSET (MORE 20)                                                            00023900
  CSET (WORDS (WHAT TELL MANY THEY THESE YOU PROGRAMS ABSTRACT HELP          00024000
```

```
  DATABASE OTHERS OR AND NOR ALL SOME ANY NO NOT NON ? .))                    00024100
EVAL ((CSETQ WORDS (CONS *DOT* WORDS)))                                       00024200
DEFINE (((EXOR (LAMBDA (X Y) (OR (AND X (NOT Y)) (AND (NOT X) Y))))))        00024300
                                                                              00024400
DEFINE                                                                        00024400
(((CAN (LAMBDA NIL (PRINL (GEXP (YOU MAY ASK QUESTIONS CONCERNING             00024500
        KEYWORDS, LANGUAGES, MANUFACTURERS, AND COMPUTERS ASSOCIATED          00024600
        WITH STORED PROGRAMS. PLEASE KEEP YOUR QUERIES SHORT AND              00024700
        SIMPLE. IF YOU ARE NOT SURE OF A CERTAIN KEYWORD-- ASK BEFORE         00024800
        YOU USE IT. THE DATABASE IS LARGE SO WATCH YOUR USE OF THE WORD       00024900
        "ALL". THE NUMBER OF ITEMS PRINTED WITH "MORE?" AND "CONTINUE?"       00025000
        MAY BE CHANGED BY TYPING "MORE=N." OR "CONTINUE=N." WHERE N IS        00025100
        A POSITIVE INTEGER.)))))))                                            00025200
                                                                              00025300
DEFINE                                                                        00025400
(((KICK (LAMBDA (IT)                                                          00025500
    (PROG (ITT ITTT FLAG)                                                     00025600
      (COND ((AND (CDR IT) (EQ (CADR IT) (QUOTE :))) (SETQ FLAG T)))          00025700
      A (COND                                                                 00025800
        ((GREATERP (LENGTH IT) (COND (FLAG (PLUS 2 MORE)) (T MORE)))          00025900
         (GO B)))                                                             00026000
      (PRINL IT)                                                              00026100
      (RETURN)                                                                00026200
      B (SETQ ITT (NCDR IT (COND (FLAG (PROGN (SETQ FLAG NIL)                 00026300
          (ADD1 MORE)))                                                       00026400
        (T (SUB1 MORE)))))                                                    00026500
      (SETQ ITTT (CDR ITT))                                                   00026600
      (RPLACD ITT NIL)                                                        00026700
      (PRINL IT)                                                              00026800
      (RPLACD ITT ITTT)                                                       00026900
      (COND ((NULL (QUSER (QUOTE MORE?) T)) (RETURN)))                        00027000
      (SETQ IT ITTT)                                                          00027100
      (GO A))))))                                                             00027200
DEFINE                                                                        00027300
(((ACHECK (LAMBDA NIL (PROG ((ATTX ATT) (NEGX NEG) INT)                       00027400
      (SETQ INT (GETPROP (CAR ATTX)))                                         00027500
      (COND ((CAR NEGX) (SETQ INT (DELETEL INT (COPY1 OBJECTS)))))            00027600
      A (SETQ ATTX (CDR ATTX))                                                00027700
      (SETQ NEGX (CDR NEGX))                                                  00027800
      (COND ((NULL ATTX) (RETURN INT)))                                       00027900
      (SETQ INT (INTERSECTION1 INT (COND ((CAR NEGX)                          00028000
          (DELETEL (GETPROP (CAR ATTX)) (COPY1 OBJECTS)))                     00028100
        (T (GETPROP (CAR ATTX))))))                                           00028200
      (GO A))))))                                                             00028300
DEFINE                                                                        00028400
(((OCHECK (LAMBDA NIL (PROG ((ATTX ATT) (NEGX NEG) UN)                        00028500
      A (COND ((NULL ATTX) (RETURN UN)))                                      00028600
      (SETQ UN (UNION1 UN (COND ((CAR NEGX)                                   00028700
          (DELETEL (GETPROP (CAR ATTX)) (COPY1 OBJECTS)))                     00028800
        (T (GETPROP (CAR ATTX))))))                                           00028900
      (SETQ ATTX (CDR ATTX))                                                  00029000
      (SETQ NEGX (CDR NEGX))                                                  00029100
      (GO A))))))                                                             00029200
DEFINE                                                                        00029300
(((SENTENCE (LAMBDA (WORD)                                                    00029400
    (COND ((NUMBERP WORD) (PRINL (GEXP (" (V. WORD) " IGNORED.))))            00029500
      ((OR (MEMB WORD OBJECTS)                                                00029600
        (MEMB WORD WORDS)                                                     00029700
        (MEMB WORD GROUPS)                                                    00029800
        (MEMB WORD KEYWORDS)                                                  00029900
        (AND (MEMB WORD MANS) (SETPROP (QUOTE MANS) T))                       00030000
        (AND (MEMB WORD AUTHORS) (SETPROP (QUOTE AUTHORS) T))
```

```
        (AND (SETQ TEMP (MEMB WORD SYNLIST))                                    00030100
             (SETQ WORD (GETPROP (CAR TEMP)))))                                 00030200
        (SETQ SENT (CONS WORD SENT)))                                           00030300
       ((NODEP WORD) (EVAL WORD))                                               00030400
       (T NIL))))))                                                             00030500
DEFINE                                                                          00030600
(((EXTGRP (LAMBDA (WORD)                                                        00030700
    (COND ((MEMB WORD GROUPS) (SETQ GRP (CONS WORD GRP))) (T NIL))))))          00030800
DEFINE                                                                          00030900
(((EXTOBJ (LAMBDA (WORD)                                                        00031000
    (COND ((MEMB WORD OBJECTS) (SETQ OBJ (CONS WORD OBJ))) (T NIL))))))         00031100
DEFINE                                                                          00031200
(((EXTATR (LAMBDA (S)                                                           00031300
    (PROG NIL A (COND ((NULL S) (RETURN))                                       00031400
      ((NOT (MEMB (CAR S) KEYWORDS)) (GO B)))                                   00031500
     (SETQ ATT (CONS (CAR S) ATT))                                              00031600
     (SETQ NEG (CONS NIL NEG))                                                  00031700
     (SETQ S (CDR S))                                                           00031800
     (COND ((NULL S) (RETURN))                                                  00031900
      ((OR (EQ (CAR S) (QUOTE NOT)) (EQ (CAR S) (QUOTE NON)))                   00032000
       (RPLACA NEG T))                                                          00032100
      (T (GO A)))                                                               00032200
     B (SETQ S (CDR S))                                                         00032300
     (GO A))))))                                                                00032400
DEFINE                                                                          00032500
(((GRPPUT (LAMBDA (ANSLIST OPTION)                                              00032600
    (PROG (IT M)                                                                00032700
     (COND ((NOT (EQ 1 (LENGTH ANSLIST))) (GO A)))                              00032800
     (SETQ IT (GETTHEM (CAR ANSLIST) (NOT (EQ 2 OPTION))))                      00032900
     (COND ((EQ 3 OPTION) (PROGN (PRINL (LENGTH IT)) (RETURN)))                 00033000
      ((NULL IT) (SETQ IT (GEXP (NONE STORED.)))))                              00033100
     (COND ((EQ 0 OPTION) (KICK IT)) (T (PRINL IT)))                            00033200
     (RETURN)                                                                   00033300
     A (SETQ M CONTINUE)                                                        00033400
     B (COND ((GREATERP M 0) NIL)                                               00033500
      ((QUSER (QUOTE CONTINUE?) T) (GO A))                                      00033600
      (T (RETURN)))                                                             00033700
     (SETQ IT (GETTHEM (CAR ANSLIST) (NOT (EQ 2 OPTION))))                      00033800
     (COND ((EQ 3 OPTION) (SETQ IT (LIST (LENGTH IT))))                         00033900
      ((NULL IT) (SETQ IT (GEXP (NONE STORED.)))))                              00034000
     (SETQ IT (CONS (CAR ANSLIST) (CONS (QUOTE :) IT)))                         00034100
     (COND ((EQ 0 OPTION) (KICK IT)) (T (PRINL IT)))                            00034200
     (COND ((NULL (SETQ ANSLIST (CDR ANSLIST))) (RETURN)))                      00034300
     (SETQ M (SUB1 M))                                                          00034400
     (GO B))))))                                                                00034500
DEFINE                                                                          00034600
(((GETTHEM (LAMBDA (C S)                                                        00034700
    (PROG ((GX GRP) PUTS GXX)                                                   00034800
     A (COND ((NULL GX) (GO C)))                                                00034900
     (SETQ GXX (CAR GX))                                                        00035000
     B (COND ((MEMB C (GETPROP (CAR GXX)))                                      00035100
       (SETQ PUTS (CONS (CAR GXX) PUTS))))                                      00035200
     (SETQ GXX (CDR GXX))                                                       00035300
     (COND (GXX (GO B)))                                                        00035400
     (SETQ GX (CDR GX))                                                         00035500
     (COND ((OR S (LESSP (LENGTH PUTS) 3)) (GO A)))                             00035600
     C (RETURN (DREVERSE PUTS)))))))                                            00035700
DEFINE                                                                          00035800
(((SHOVEABS (LAMBDA (L)                                                         00035900
    (PRINL (GEXP (ABSTRACTS NOT AVAILABLE-- ASK FOR KEYWORDS.))))))))           00036000
```

```
DEFINE                                                                  00036100
(((CONTROL (LAMBDA NIL (PROG (ERM)                                      00036200
      (PRINL (GEXP (ENQUIRE: ENGLISH QUESTION INTERPRETATION AND        00036300
        RESPONSE.)))                                                    00036400
      (PRINL (GEXP (DATABASE: COMPUTER PROGRAMS.)))                     00036500
      (SETQ (SYMSWITCH . 124) (SETQ RESULT NIL))                        00036600
      A (SETQ SENT (SETQ OBJ (SETQ NEG (SETQ ATT (SETQ GRP NIL)))))    00036700
      TRY (TRY ERM B (ENQUIRE))                                         00036800
      (GO A)                                                            00036900
      B (COND (ERRORS (PRINT ERM)))                                     00037000
      (PRINL (QUOTE ????))                                              00037100
      (GO A)))))))                                                      00037200
(ENQC EVAL                                                              00037300
((CSETQ ANAMESSAGE (GEXP (ANAPHORIC REFERENCE WORDS $$@(THEY,@ THEM,    00037400
    $$@ETC.)@ CAN ONLY REFER TO OBJECTS $$@(PROGRAM@ $$@NAMES)@ IN THE 00037500
    DATABASE.))))                                                       00037600
DEFINE                                                                  00037700
(((ENQUIRE (LAMBDA NIL (PROG (BS SENT1 ANS ANS1 ANS3 A C N TH W L D TL 00037800
      O LL Q)                                                           00037900
      (SETQ SENT1 (SCANNER (CONREAD)))                                  00038000
      (COND ((AND (EQ (LENGTH SENT1) 4) (EQ (CADR SENT1) (QUOTE =)))    00038100
        (PROGN (CSET (CAR SENT1) ((EVAL . 122) (CADDR SENT1)))          00038200
        (PRINL (QUOTE SET.))                                            00038300
        (RETURN)))                                                      00038400
       ((SETQ TEMP (INTERSECTION1 SENT1 BLOCKERS))                      00038500
        (PROGN (PRINL (GEXP (" (V. TEMP) " ILLEGAL.))) (SETQ BS T))))   00038600
      (COND ((SETQ TEMP (INTERSECTION1 SENT1 STOPPERS))                 00038700
        (PROGN (PRINL (GEXP (NO INFORMATION ON " (V. TEMP)              00038800
            " IN DATABASE.)))                                           00038900
        (SETQ BS T))))                                                  00039000
      (COND (BS (GO RETURN))                                            00039100
       ((MEMB (QUOTE GOODBYE) SENT1)                                    00039200
        (PROGN (PRINL (GEXP (SEE YOU SOON.)))                           00039300
        (CODE (SVC 0))                                                  00039400
        (PRINL (GEXP (HELLO AGAIN.)))                                   00039500
        (RETURN)))                                                      00039600
       ((OR (EQUAL SENT1 (QUOTE (?)))                                   00039700
        (MEMB (QUOTE USER) SENT1)                                       00039800
        (MEMB (QUOTE HELP) SENT1))                                      00039900
        (PROGN (CAN) (GO RET))))                                        00040000
      (MAPCAR SENT1 (FUNCTION SENTENCE))                                00040100
      (SETQ SENT (CDR SENT))                                            00040200
      (COND ((NULL SENT) (GO FAIL)))                                    00040300
      (SETQ SENT (INTERSECTION1 SENT SENT))                             00040400
      (COND ((MEMB (QUOTE DATABASE) SENT) (SETQ D T)))                  00040500
      (COND ((AND (NOT D) (SETQ TEMP (MEMB (QUOTE YOU) SENT)))         00040600
        (RPLACA TEMP (QUOTE ENQUIRE))))                                 00040700
      (COND ((MEMB (QUOTE WHAT) SENT) (SETQ W 1)))                      00040800
      (COND ((MEMB (QUOTE TELL) SENT)                                   00040900
        (COND (W (GO SIMPLIFY)) (T (SETQ W (SETQ TL 1))))))             00041000
      (COND ((MEMB (QUOTE MANY) SENT)                                   00041100
        (COND (W (GO SIMPLIFY)) (T (SETQ W 2)))))                       00041200
      (COND ((MEMB (QUOTE THEY) SENT) (SETQ TH 1)))                     00041300
      (COND ((MEMB (QUOTE THESE) SENT) (SETQ TH 2)))                    00041400
      (COND ((MEMB (QUOTE PROGRAMS) SENT) (SETQ C T)))                  00041500
      (COND ((MEMB (QUOTE ABSTRACT) SENT) (SETQ A T)))                  00041600
      (COND ((MEMB (QUOTE OTHERS) SENT) (SETQ O T)))                    00041700
      (COND ((MEMB (QUOTE OR) SENT) (SETQ L 1)))                        00041800
      (COND ((MEMB (QUOTE AND) SENT)                                    00041900
        (COND (L (GO SIMPLIFY)) (T (SETQ L 2)))))                       00042000
```

```
(COND ((MEMB (QUOTE NOR) SENT)                                          00042100
  (COND (L (GO SIMPLIFY)) (T (SETQ L 3)))))                             00042200
(COND ((MEMB (QUOTE ALL) SENT) (SETQ Q 1)))                             00042300
(COND ((MEMB (QUOTE ANY) SENT)                                          00042400
  (COND (Q (GO SIMPLIFY)) (T (SETQ Q 2)))))                             00042500
(COND ((MEMB (QUOTE SOME) SENT)                                         00042600
  (COND (Q (GO SIMPLIFY)) (T (SETQ Q 3)))))                             00042700
(COND ((MEMB (QUOTE NO) SENT)                                           00042800
  (COND (Q (GO SIMPLIFY)) (T (SETQ Q 4)))))                             00042900
(COND ((MEMB (QUOTE NOT) SENT) (SETQ N T)))                             00043000
(MAPCAR SENT (FUNCTION EXTOBJ))                                         00043100
(MAPCAR SENT (FUNCTION EXTGRP))                                         00043200
(EXTATR SENT)                                                           00043300
(COND ((AND GRP A) (GO SIMPLIFY))                                       00043400
 (GRP (SETQ GRP (MAPCAR GRP (FUNCTION (EVAL . 122))))))                 00043500
PROCESS (COND ((AND (EQ 2 TH) (EQ 1 (LENGTH RESULT)))                   00043600
  (SETQ TH 1)))                                                         00043700
(COND (O (COND (OBJ (SETQ ANS (DELETEL OBJ (COPY1 OBJECTS))))         00043800
   (RESULT (SETQ ANS (DELETEL (LAST RESULT) (COPY1 OBJECTS))))         00043900
   (T (GO FAIL))))                                                      00044000
 (OBJ (SETQ ANS OBJ))                                                   00044100
 ((EQ 1 TH)                                                             00044200
  (COND ((NULL RESULT) (GO ANAFAIL))                                    00044300
   (T (SETQ ANS (COPY1 (LAST RESULT))))))                               00044400
 ((EQ 2 TH)                                                             00044500
  (COND ((NULL RESULT) (GO ANAFAIL))                                    00044600
   (T (SETQ ANS (COPY1 (CAR RESULT))))))                                00044700
 (C (SETQ ANS (COPY1 OBJECTS))))                                        00044800
(COND (ANS (GO P2))                                                     00044900
 (ATT (GO P1))                                                          00045000
 ((NULL GRP) (GO P0.5))                                                 00045100
 ((EQ 2 W) (PRINL (MAPCAR GRP (FUNCTION LENGTH))))                      00045200
 ((NULL W)                                                              00045300
  (COND ((NOT (EQ 1 (LENGTH GRP))) (GO SIMPLIFY))                       00045400
   (T (PRINL (COND ((EQ 2 Q) (QUOTE YES.)) (T (QUOTE NO.))))))))       00045500
 ((OR (NULL Q) (EQ 2 Q) (EQ 3 Q)) (MAPCAR GRP (FUNCTION KICK)))        00045600
 ((EQ 1 Q) (MAPCAR GRP (FUNCTION PRINL)))                               00045700
 (T (GO FAIL)))                                                         00045800
(GO RET)                                                                00045900
P0.5 (COND ((OR A (NULL D)) (GO FAIL))                                  00046000
 ((EQ 2 W) (PRINL (QUOTE LOTS.)))                                       00046100
 ((EQ 1 W) (CAN))                                                       00046200
 (T (GO FAIL)))                                                         00046300
(GO RET)                                                                00046400
P1 (COND (GRP (PROGN (PRINL (COND ((MEMB (CAR ATT) (CAR GRP))          00046500
       (QUOTE YES.))                                                    00046600
      (T (QUOTE NO.))))                                                 00046700
   (GO RET)))                                                           00046800
 ((NULL A) (GO FAIL)))                                                  00046900
(SETQ ANS (COPY1 OBJECTS))                                              00047000
(GO P4)                                                                 00047100
P2 (COND (ATT (GO P4))                                                  00047200
 (GRP (GO P3))                                                          00047300
 (A (SHOVEABS ANS))                                                     00047400
 ((EQ 2 W) (PRINL (LENGTH ANS)))                                        00047500
 ((EQ 1 W)                                                              00047600
  (COND ((NULL Q)                                                       00047700
   (COND ((AND (LESSP (LENGTH ANS) 4) (OR TL D))                        00047800
     (SHOVEABS ANS))                                                    00047900
    (T (KICK ANS))))                                                    00048000
```

```
      ((EQ 1 Q) (PRINL ANS))                                                    00048100
      ((EQ 3 Q)                                                                 00048200
       (PRINL (COND ((EQ 1 (LENGTH ANS))                                        00048300
          (GEXP ((V. (CAR ANS)) (IS THE ONLY ONE.))))                           00048400
         ((EQ 2 (LENGTH ANS))                                                   00048500
          (GEXP ((V. (CAR ANS))                                                 00048600
            AND (V. (CADR ANS))                                                 00048700
            (ARE THE ONLY ONES.))))                                             00048800
         (T (SETQ ANS (LIST (CAR ANS) (CADR ANS)))))))                          00048900
      ((EQ 2 Q) (PRINL (SETQ ANS (LAST ANS))))                                  00049000
      (T (PRINL (QUOTE ???))))))                                                00049100
     (T (PRINL (COND ((EQ (LENGTH ANS) (LENGTH OBJECTS))                        00049200
       (COND ((OR (EQ 2 Q) (EQ 3 Q)) (QUOTE YES.))                              00049300
        (T (QUOTE NO.))))                                                       00049400
      ((EQ 1 L) (GEXP (PLEASE SPLIT YOUR REQUESTS.)))                           00049500
      (C (COND (N (QUOTE NO.)) (T (QUOTE YES.))))                               00049600
      (T (QUOTE ???)))))))                                                      00049700
.(GO RET1)                                                                      00049800
P3 (COND (W (GRPPUT ANS (COND ((EQ 1 W)                                         00049900
      (COND ((OR (EQ 2 Q) (EQ 3 Q)) 2) ((EQ 1 Q) 1) (T 0)))                    00050000
      (T 3))))                                                                  00050100
     (T (PRINL (QUOTE NO.))))                                                   00050200
(GO RET1)                                                                       00050300
P4 (COND ((AND (EQ 1 (LENGTH ATT)) (EQ 1 L)) (GO FAIL))                         00050400
     ((EQ 1 L) (SETQ LL T))                                                     00050500
     ((EQ 3 L)                                                                  00050600
      (SETQ NEG (MAPCAR NEG (FUNCTION (LAMBDA (X) (NOT X)))))))                 00050700
(COND ((MEMB T NEG) (PRINL (GEXP (ONE MOMENT PLEASE.)))))                       00050800
(SETQ ANS1 (INTERSECTION1 ANS (COND (LL (OCHECK)) (T (ACHECK)))))               00050900
(SETQ ANS3 (COND (ANS1 ANS1) (T ANS)))                                          00051000
(COND ((EQ 1 TH)                                                                00051100
       (COND ((EQ (LENGTH RESULT) 2) (RPLACA RESULT ANS3))                      00051200
        (T (SETQ RESULT (CONS ANS3 RESULT)))))                                  00051300
     (T (SETQ RESULT (LIST ANS3))))                                             00051400
(COND (GRP (GO P5))                                                             00051500
     (A (SHOVEABS ANS1))                                                        00051600
     ((EQ 1 W)                                                                  00051700
      (COND ((NULL ANS1) (PRINL (GEXP (THERE ARE NONE.))))                      00051800
       ((EQ 1 Q) (PRINL ANS1))                                                  00051900
       (T (KICK ANS1))))                                                        00052000
     ((EQ 2 W) (PRINL (LENGTH ANS1)))                                           00052100
     (T (PRINL (COND ((OR (EQ 2 Q) (EQ 3 Q))                                    00052200
         (COND (ANS1 (QUOTE YES.)) (T (QUOTE NO.))))                            00052300
        ((EQ (LENGTH ANS1) (LENGTH ANS)) (QUOTE YES.))                          00052400
        (T (QUOTE NO.))))))                                                     00052500
(GO RETURN)                                                                     00052600
P5 (COND (ANS1 (GRPPUT ANS1 (COND ((OR (NULL W) (EQ 2 W)) 3)                    00052700
       ((OR (EQ 2 Q) (EQ 3 Q)) 2)                                               00052800
       ((EQ 1 Q) 1)                                                             00052900
       (T 0))))                                                                 00053000
     (T (PRINL (GEXP (NO SUCH $$@PROGRAM(S).@)))))                              00053100
(GO RETURN)                                                                     00053200
SIMPLIFY (PRINL (GEXP (PLEASE SIMPLIFY YOUR QUESTION.)))                        00053300
(GO RETURN)                                                                     00053400
FAIL (PRINL (QUOTE ???))                                                        00053500
(GO RETURN)                                                                     00053600
ANAFAIL (PRINL ANAMESSAGE)                                                      00053700
(GO RETURN)                                                                     00053800
RET (SETQ RESULT NIL)                                                           00053900
(GO RETURN)                                                                     00054000
```

```
        RET1 (SETQ RESULT (LIST ANS))                                          00054100
        RETURN (RETURN)))))))                                                  00054200
  (ENQD SPECIAL ((KEYWORDS OBJECTS GROUPS SYNLIST))                            00054300
  DEFINE (((CONS- (LAMBDA (L) (CONS *MINUS* L)))))                             00054400
  DEFINE                                                                       00054500
  (((ATOMIZE (LAMBDA (L)                                                       00054600
       (COND ((ATOM L) L)                                                      00054700
        (T (COMPRESS (CDR (UNLIST (MAPCAR (MAPCAR L (FUNCTION EXPLODE))       00054800
             (FUNCTION CONS-)))))))))))                                       00054900
  DEFINE                                                                       00055000
  (((DRGEN (LAMBDA (Q)                                                         00055100
       (PROG (IT WORD ITEM X NAMES (COUNT 0))                                  00055200
        (OPEN1 Q)                                                              00055300
        (SETQ IT (RDS (CAR Q)))                                                00055400
        Z (COND ((EQ (SETQ X (READ)) (QUOTE EOT)) (GO W)))                     00055500
        (SETQ NAMES (CONS X NAMES))                                            00055600
        (GO Z)                                                                 00055700
        W (SETQ NAMES (DREVERSE NAMES))                                        00055800
        (SETQ WORD (READ))                                                     00055900
        A (SETQ COUNT (ADD1 COUNT))                                            00056000
        (COND ((LESSP COUNT 100) (GO AA)))                                     00056100
        (PRINT *MINUS*)                                                        00056200
        (SETQ COUNT 0)                                                         00056300
        AA (COND ((EQ WORD (QUOTE EOF)) (GO E)))                               00056400
        (SETQ WORD (ATOMIZE (CAR WORD)))                                       00056500
        (SETQ KEYWORDS (CONS WORD KEYWORDS))                                   00056600
        (SETQ ITEM (READ))                                                     00056700
        B (COND ((OR (NULL ITEM) (NULL (CAR ITEM))) (GO C))                    00056800
         ((EQ WORD (SETQ X (ATOMIZE (CAR ITEM)))) (GO Y)))                     00056900
        (SETPROP (CAR (SETQ SYNLIST (CONS X SYNLIST))) WORD)                   00057000
        Y (SETQ ITEM (CDR ITEM))                                               00057100
        (GO B)                                                                 00057200
        C (READ)                                                               00057300
        D (COND ((NOT (NUMBERP (SETQ ITEM (READ))))                            00057400
          (PROGN (SETQ WORD ITEM) (GO A))))                                    00057500
        (SETQ ITEM (CADR (MEMB ITEM NAMES)))                                   00057600
        (COND ((MEMB ITEM (GETPROP WORD)) (GO D)))                             00057700
        (SETPROP WORD (CONS ITEM (GETPROP WORD)))                              00057800
        (GO D)                                                                 00057900
        E (RDS IT)                                                             00058000
        (SHUT1 Q)                                                              00058100
        (SETQ KEYWORDS (DREVERSE KEYWORDS))                                    00058200
        F (COND ((NULL NAMES) (GO G))                                          00058300
         ((NUMBERP (CAR NAMES)) NIL)                                           00058400
         (T (SETQ OBJECTS (CONS (CAR NAMES) OBJECTS))))                        00058500
        (SETQ NAMES (CDR NAMES))                                               00058600
        (GO F)                                                                 00058700
        G (SETQ OBJECTS (SORT OBJECTS))                                        00058800
        (RETURN (QUOTE EOF)))))))                                              00058900
  DEFINE                                                                       00059000
  (((DUMP (LAMBDA (Q1 Q2)                                                      00059100
       (PROG (IT1 IT2 X Y)                                                     00059200
        (OPEN1 Q1)                                                             00059300
        (SETQ IT1 (RDS (CAR Q1)))                                              00059400
        (OPEN1 Q2)                                                             00059500
        (SETQ IT2 (WRS (CAR Q2)))                                              00059600
        B (COND ((NOT (EQ (READ) (QUOTE EOT))) (GO B)))                        00059700
        A (COND ((EQ (SETQ X (READ)) (QUOTE EOF)) (GO C))                      00059800
         ((NUMBERP X) (GO A)))                                                 00059900
        (PRINL (INTERSECTION1 (SETQ Y (CONS (ATOMIZE (CAR X))                  00060000
```

```
          (MAPCAR (READ) (FUNCTION ATOMIZE))))                                  00060100
        Y))                                                                     00060200
       (READ)                                                                   00060300
       (GO A)                                                                   00060400
       C (SHUT1 Q1)                                                             00060500
       (RDS IT1)                                                                00060600
       (SHUT1 Q2)                                                               00060700
       (WRS IT2)                                                                00060800
       (RETURN (QUOTE EOF)))))))                                                00060900
(ENQX CSET (GROUPS (KEYWORDS LANGUAGES COMPUTERS AUTHORS MANS))                 00061000
 CSET (COMPUTERS (BURROUGHS5500 CDC1604 CDC3400 CDC3600 CDC3600-COMPASS         00061100
   CDC6000 CDC6400 CDC6500 CDC6600 DEC-PDP10 GE635 IBM1130 IBM1401              00061200
   IBM1620 IBM1802 IBM360 IBM360/40 IBM360/50 IBM360/65 IBM360/67               00061300
   IBM360/91 IBM370/145 IBM7090 IBM7040 IBM709 IBM7094 RCA-SPECTRA70            00061400
   RCA/2-7 SPECTRA70 UNIVAC-SDF UNIVAC1108 UNIVAC1108-ASSEMBLER                 00061500
   UNIVAC1108-EXEC8))                                                           00061600
 CSET (LANGUAGES (ALGOL ASSEMBLER BAL BASIC PL/1 SNOBOL4 SPITBOL                00061700
   FORTRAN FORTRAN-II FORTRAN-IV FORTRAN-V FORTRAN-63 LISP WATFIV               00061800
   WATFOR WATIV))                                                               00061900
 CSET (MANS (IBM CDC DEC RCA BURROUGHS UNIVAC CALCOMP GE))                      00062000
 CSET (AUTHORS (LEAL))                                                          00062100
 (LAMBDA NIL (PROG NIL (CSETQ SYNLIST (APPEND (GEXP (AUTHOR WROTE NAME          00062200
       RUN COUNT WHICH LIST PRINT GIVE WHO WHOM THEM THEIR ONE ONES IT          00062300
       ITS THOSE YOUR YOURSELF PROGRAM ABSTRACTS MEMORY INFORMATION             00062400
       MANUFACTURERS COMPUTER MACHINE MANUFACTURER MANUFACTURE                  00062500
       MANUFACTURES KNOWLEDGE KNOW OTHER EXCEPT & KEYWORD PROPERTY              00062600
       PROPERTIES LANGUAGE))                                                    00062700
    SYNLIST))                                                                   00062800
   (SETPROP (QUOTE LIST) (QUOTE WHAT))                                          00062900
   (SETPROP (QUOTE NAME) (QUOTE WHAT))                                          00063000
   (SETPROP (QUOTE RUN) (QUOTE COMPUTERS))                                      00063100
   (SETPROP (QUOTE COUNT) (QUOTE MANY))                                         00063200
   (SETPROP (QUOTE YOUR) (QUOTE YOU))                                           00063300
   (SETPROP (QUOTE YOURSELF) (QUOTE YOU))                                       00063400
   (SETPROP (QUOTE PROPERTY) (QUOTE KEYWORDS))                                  00063500
   (SETPROP (QUOTE PROPERTIES) (QUOTE KEYWORDS))                                00063600
   (SETPROP (QUOTE COMPUTER) (QUOTE COMPUTERS))                                 00063700
   (SETPROP (QUOTE MACHINE) (QUOTE COMPUTERS))                                  00063800
   (SETPROP (QUOTE MANUFACTURER) (QUOTE MANS))                                  00063900
   (SETPROP (QUOTE MANUFACTURERS) (QUOTE MANS))                                 00064000
   (SETPROP (QUOTE MANUFACTURE) (QUOTE MANS))                                   00064100
   (SETPROP (QUOTE MANUFACTURES) (QUOTE MANS))                                  00064200
   (SETPROP (QUOTE WHICH) (QUOTE WHAT))                                         00064300
   (SETPROP (QUOTE PRINT) (QUOTE WHAT))                                         00064400
   (SETPROP (QUOTE GIVE) (QUOTE WHAT))                                          00064500
   (SETPROP (QUOTE WHO) (QUOTE WHAT))                                           00064600
   (SETPROP (QUOTE WHOM) (QUOTE WHAT))                                          00064700
   (SETPROP (QUOTE THEM) (QUOTE THEY))                                          00064800
   (SETPROP (QUOTE THEIR) (QUOTE THEY))                                         00064900
   (SETPROP (QUOTE ONE) (QUOTE THEY))                                           00065000
   (SETPROP (QUOTE ONES) (QUOTE THEY))                                          00065100
   (SETPROP (QUOTE IT) (QUOTE THEY))                                            00065200
   (SETPROP (QUOTE ITS) (QUOTE THEY))                                           00065300
   (SETPROP (QUOTE THOSE) (QUOTE THESE))                                        00065400
   (SETPROP (QUOTE PROGRAM) (QUOTE PROGRAMS))                                   00065500
   (SETPROP (QUOTE ABSTRACTS) (QUOTE ABSTRACT))                                 00065600
   (SETPROP (QUOTE MEMORY) (QUOTE DATABASE))                                    00065700
   (SETPROP (QUOTE INFORMATION) (QUOTE DATABASE))                               00065800
   (SETPROP (QUOTE KNOWLEDGE) (QUOTE DATABASE))                                 00065900
   (SETPROP (QUOTE KNOW) (QUOTE DATABASE))                                      00066000
```

```
    (SETPROP (QUOTE OTHER) (QUOTE OTHERS))                          00066100
    (SETPROP (QUOTE EXCEPT) (QUOTE OTHERS))                         00066200
    (SETPROP (QUOTE &) (QUOTE AND))                                 00066300
    (SETPROP (QUOTE KEYWORD) (QUOTE KEYWORDS))                      00066400
    (SETPROP (QUOTE LANGUAGE) (QUOTE LANGUAGES))                    00066500
    (SETPROP (QUOTE AUTHOR) (QUOTE AUTHORS))                        00066600
    (SETPROP (QUOTE WROTE) (QUOTE AUTHORS))                         00066700
    (RETURN (QUOTE SYNLIST))))                                      00066800
NIL)                                                                00066900
```