Check for updates

SYMBOLIC LOGIC IN LANGUAGE ENGINEERING

H. M. Semarne

Research Programming Coordinator
Douglas Aircraft Company, Inc.
Santa Monica, California

Introduction

Not long ago, information activities were still considered merely incidental to the technical operations of industrial concerns, while such activities are now fast becoming a tangible portion of company efforts. The problems of information storage and retrieval increase even while solution methods are being studied. There is an urgent need for new techniques which will enable clerical help with machines to carry out searches, thus liberating engineers and scientists for work on technical problems. These, in turn, include the problems of documentation research.

In order to organize the use of recorded knowledge, it is obviously desirable to:

1. Classify all information to be stored;

2. Correlate all facts to guide future decisions;

3. Select recorded information relative to a given problem.

The development of a large-scale retrieval system will largely depend on the results of work in the classification of concepts rather than of words. There is little sense in classifying words to organize information because the irrational, natural language of a document is hardly ever the best systematic language for use in a machine. This difficulty is a serious obstacle to the accurate correlation of recorded facts and, by extension, to phrase-by-phrase translation from one natural language into another. Since natural languages are rather inefficient tools for the symbolic expression of ideas, the methods of symbolic logic can be employed to analyze, formulate, and correlate these ideas before they are used, "couched in the capricious imagery of words", so to say.

Making use of the symbolic logic truth-matrix computer technique discussed in earlier papers 1, 2, and of other logic methods, it is possible to construct decision-devices useful in the classification, correlation, and selection of documents, as well as in the machine translation of languages. The basis of the symbolic logic approach is non-arithmetical. It features a very flexible matching ability. The approach, general with respect to machinery, does not suffer from the handicap of a searching program written expressly for an existing special-purpose machine. The latter may operate by quasi-mathematical processes, and entail long stretches of "housekeeping" operations.

Another important consideration in the planning of an information retrieval system is its adaptability to expansion. It is one of the purposes of the symbolic logic methods to provide for the eventual embedding of the present system in one of far greater scope, preferably in nesting-block fashion.

The approach taken is not to be confused with one limited to the use of the logic of collections of messages without regard for meaning. It is here intended to do more than merely catalog the frequencies of musical tones in hopes of having someone perceive a melody.

The Progressive Stages Of Logic Analysis

The techniques of mathematical or symbolic logic which underlie the proposed approach to documentation research will be discussed only insofar as they go beyond the rudiments of this discipline as given in textbooks on symbolic logic or, in part, in 1, 2. It is the object of the present paper to show how symbolic logic, as a tool, can be introduced gradually into the problems of information storage and machine searching and translation, just as the scope and the flexibility of the system developed are gradually increased.

1. Seven two-propositional functors and denial (Figure 1) are used to analyze the relationships of a body of information and to separate out various classes of factors. The relationships thus brought out might go unnoticed in the confusing form in which information, such as a large group of observational data, or a set of statements in a natural language, is usually available. For instance, a chemist may encounter this:

   A white crystalline solid contains nitrogen. Its melting point is 114° and the solubility division is W. The aqueous solution is neutral to litmus. When the solid is heated, it distils at 222°; water is evolved during this distillation. The distillate reacts with nitrous acid to form a water soluble acid that boils at 118°.

A logic statement corresponding to this description may be given as: (see Figure 2 for proposition-key)

KKKNaKcNdCcPab
KNge
KKKijkl
Kqr
KPonm
Kuv
Kyz

Note that for propositions k,l,m, and z table-lookup is indicated

2. By means of the basic two-propositional functors, the truth-matrix technique of[1] and a method of encoding and matching through the characteristics (truth-values) of statements, a convenient and versatile basis is provided for searching, sorting and selecting operations. (see example below)

3. Group-theoretical relations[3] among the characteristics can be used to simplify considerably the making of decisions. Tedious operations commonly performed by Boolean algebra are thus paralleled by a binary matching technique which makes possible the ordering of otherwise very unwieldy clusters of statements. Through this technique, simplification and normalization (i.e. transcription into the form using only and, or, not) become automatic.

   All the characteristics of a given number of propositions form a finite Abelian group with respect to the operation $\wedge$. By comparing characteristics through the equivalence operation - 1 if a pair of bits is alike, 0 otherwise - it is possible to have the machine find group elements which have the simplest symbol equivalents.

   e.g., the expression

   VKVNPabcNVVNSabNSbcNSacKKabc
   has characteristic 0 1 1 0 1 0 0 1
   (Figure 3)

   Testing the characteristic of a:

0 1 1 0 1 0 0 1	characteristic
0 0 0 0 1 1 1 1	a
1 0 0 1 1 0 0 1	by E-(equivalence) operation
0 1 1 0 0 1 1 0	inverse, (by N-operation) this indicates the presence of the A-operation
0 1 1 0 0 1 1 0	$b \wedge c$
1 1 1 1 1 1 1 1	tautology proves this

   Hence, the expression simplifies to
   $a \wedge (b \wedge c)$
   or AaAbc
   or, in normal form, (since Axy is the same as VKNyxKNyx)
   VKNVKNcbKNbcaKNaVKNcbKNbc

4. Probabilistic weighting factors can be admitted to provide for the statistical handling of information. For example, they could serve for machine-decisions concerning pertinency ratings of documents retrieved.

   For example, a materials engineer may make the following request: What rubber material should be used in a certain O-ring packing exposed to organic phosphate ester hydraulic fluid on the inside surface and atmospheric oxygen on the outside surface, and able to withstand 225°F hydraulic system temperature and mechanical abrasion.

   There are perhaps 25 families of rubber materials against which to match the resulting assertion pattern. Some of the required properties may be obtained more generously than others from any given rubber material. It may be necessary to establish a preference order among disclosures, since it would be desirable to obtain not only a material just fulfilling the requirements, but the best possible one for the purpose at hand. The task may be carried out as outlined in Figure 4. This example may serve to illustrate the high degree of specification attainable in this type of machine searching.

5. In order to achieve more freedom in translating diversified and intricate statements in a natural language into machine language, logic operators,[4] or quantifiers (Figure 5) could be called in.

   These quantifying operators could be introduced as special notes by which to evaluate properly the results of a truth-matrix analysis. Alternatively, the quantifiers could be approximated, for purposes of computation, by prearranged threshold values which would then be used like probabilistic weighting factors (above).

The functional relationship of information retrieval systems to arithmetic compiler languages may be noted here. Documentation work essentially deals with patterns of thought and their relations to specific units of recorded information. While it may be undesirable to mold work in such a discipline on a technique of programming mainly arithmetical processes, it may nevertheless be profitable to uphold compilers as good examples, philosophically speaking. Compiler languages, such as Fortran, Speed, etc., are representations at the operator, or function, level. The transcription of documents, instead of being done on an elementary symbol-by-symbol basis, without reference to context, should also graduate to the function level. The relationships among the clue-words of a document should be inherent in the record.

The Progressive Stages Of Information Handling

Before proceeding with the application of symbolic logic, one may review the need for it.

The point of departure for all documentation research can be said to be the traditional library index. The first step in this work is due to the need to go from the fixed and little expandable subject categories of the library index to flexible categories of information such as are provided by many special libraries today.

In multi-aspect indexing, documents are stored under groups of clue-words which characterize their contents, and retrieval is effected by requests made up of any combination of clue-words. This is a long step ahead and away from fixed-category filing.

It now appears that the machine runs into a snare on retrieval. If all the clue-words pertinent to a document are combined into one large group, there will be false drops due to lack of resolution. If, on the other hand, a document is indexed only under a few clue-words, it may be overlooked on many requests. That is to say, much of the advantage of the system over traditional indexing is then lost.

This is the central quandary of information handling. Many authors have attacked the problem in many ingenious ways, and some of them have successfully moved far beyond that stage, [5,6,7,8,9,10] to find an optimum solution.

One trend is thus toward more detailed indexing, and another toward simplification. Moreover, these trends are also tied in with machine speeds and machine memory capacities. Consequently, most of the new work in documentation research is centered at the intersection of these two trends. Among the varied results which seem to arise, not all, unfortunately, are the products of the essential double-criterion:

Is the system commensurate with the specific information needs to be dealt with, and is it extensible?

Three levels of document description can be defined:

1. identification of symbols
2. identification of their order
3. identification of their logic relations

In this last level, information concepts can find expression through logic relations among the underlying clue-words. (Figure 6)

Symbolic Logic Treatment Of Information

Machines are now linked with the storage, analysis, and retrieval of information. There is little point in attempting to discuss information handling without rapid electronic computers, but it must be realized that the greatest contribution of the machine to this field does not lie in speed, but in the right type of searching.

The latter depends on the design of the search program which depends, in part, on the method of storing the information. From the viewpoint of logic, the encoding of the information for storage and the encoding of search requests can be regarded as one and the same task. The mechanics of storage assignment for the former, and the mechanics of matching clues for the latter operation, are the only important points in which they are different.

To write a search request means to write a set of specifications which should be as precise as possible. The burden on the writer of the request can be immensely lightened and he can be enabled to write a great many such requests in the time formerly allotted to one request, by the simple expedient of having the machine compose the request for him. This can be done through symbolic logic, as shown above (Figures 2,4,6).

All the essential facts known about a document, such as the title, serial number, origin, year of publication, location in a file, descriptive terms, and its subject content, (materials involved, properties observed, processes performed, ambient conditions, applications, etc.) are merely listed, regardless of possible redundancies. Each one of the facts or propositions listed may be linked with some of the other propositions or with extraneous useful matter concerning the document sought. There now arise a number of statements, such as: a or else b, (Aab) a or b or c or all three, (VVabc), a implies c but not b unless d and e are present, (KCVNdNeCacCKde CaKcb), etc. The machine is directed to run through the Truth-Matrix Analysis as discussed earlier[1], and the result is a binary number of 2^n bits (for n propositions). This number, called the characteristic of the statement, is obtained one bit at a time; for every "yes" or 1 bit, a binary number of n bits (for n propositions) denoting the combination of propositional values corresponding to the "yes" result, is automatically recorded in memory. A set of these numbers, corresponding to a given document, constitutes a complete assertion pattern for that document (on encoding) or a complete description of the search request (on retrieval). A special but frequently occurring case is that of disjoint characteristics which is due to a statement in which the propositions can be grouped in independent, or disjoint, substatements.

For the example used above, the disjoint characteristics would be:

0010000000000000
0011001100000000
0000000000001111
1111111100000000
0000000011000000
0000000000001111
0000001100000011

The assertion pattern derived from the above characteristics would be:

0010	0000	1100
	0001	1101
0010	0010	1110
0011	0011	1111
0110	0100	
0111	0101	0110
	0110	0111
1100	0111	1110
1101		1111
1110	1000	
1111	1001	

This assertion pattern is compared with assertion patterns of all entries within a pertinent range. The pertinent range is defined by combining the original statements with a standard set of conclusions, e.g. implications like: water evolution on distillation of a solid implies a hydrate, reaction with nitrous acid implies the presence of a primary or secondary amino-group etc. The resulting combined characteristic clearly outlines the possible entries sought, and the search pinpoints it.

The basic logic "in-and-out" procedure may be summarized as follows:

a. Analyzing and Coding:

1. Collect facts or propositions
2. Compose propositions into logic statement
3. Set up proposition-key (dictionary), by defining the propositions
4. Analyze statement and obtain its identifying assertion pattern
5. Store characteristic and proposition-key addresses

b. Analyzing and Retrieval:

1. Collect facts or propositions
2. Compose request propositions into logic statement
3. Analyze statement and obtain its assertion pattern
4. Match assertion patterns
5. Check proposition-key for print-out of meanings.

The disclosure obtained in this manner is highly unambiguous, having been released upon matching a assertion pattern which denotes the same logic relationships as that of the request.

The dictionary entries would be kept out of the way of the characteristics. When part of a document record is devoted to textual material and part to a code, the effective search rate is severely cut due to the idling of the circuitry during the passage of the textual part of the record. The search operation should therefore be broken into two portions, the first being the matching of the logic form or assertion pattern of the document and of the proposition code, and the second being the special table look-up in the proposition-key (Figures 2,4,6). The latter operation results in a match. On the other hand, the proposition code and its dictionary could remain related for easy identification of the code for a document.

The amount of detail used in describing concepts stored or disclosed in the document file depends on the needs of the user and the nature of the application. These will also determine whether a single-shot search procedure or an iterative analysis procedure is to be followed. If the file for an area is appreciably less branched in its descriptive terms than is the type of request addressed to it, the iterative analysis approach may be more advisable. No special encoding method is necessary in the preparation of iterative analysis searches. The form of the request suffices to determine the depth of the search and the logic classes to be considered during that search, or that step of the search. The gradual branching of logic decisions is automatically controlled by changes in the request statement.

The iterative variant of symbolic logic searching can be employed to determine the degree of relevance of a disclosure as compared with that of another. It is necessary only to run several searches, each passing through a different logic network and to count the matches of certain individual clue-words on the records of various documents examined.

Again, it is the needs and the skill of the user, and the nature of the application, that determine the extent of automaticity desired in the composition of the request sentence.

A further study of the role to be played by logic request sequences in the guiding of the design of electronic equipment for searching and selecting should be made.

Sufficient Clues And Probability Aspects

Besides the need for retrieving information, there is also the necessity to analyze documents for various purposes. It may be desired, for instance, to examine a collection of data from various sources in order to deduce the answer to an apparently extraneous question. Or, on receiving or recalling various barely related facts, it may be desired to find a special pertinent document. Such situations will be referred to as "sufficient clues" problems. (The word, sufficient, is used on a hopeful note).

Suppose, as a simple example, that these items are collected:

1. There are three authors, x,y,z, each having written one of three documents, α, β, γ, not necessarily in this order.
2. There are three other documents, a,b,c, the respective authors of which are the same as those of the three above documents, respectively.
3. Document c deals with topic one
4. Author y writes on a second topic
5. Document b is not mathematical in nature
6. One of documents a,b,c, namely the one marked by the corresponding greek letter, (designating a paper written by y) deals with a third topic.
7. Author y has written on the topic of that one among documents a,b,c which has mathematical symbols in it.
8. Author z does not know document α

Upon a question such as: Which is the author of document α? It can be shown that the straightforward answer, "x is" can be rather suggestively obtained by means of the same symbolic logic truth-matrix analysis as that used above. It is necessary merely to make a proposition-dictionary, jot down all the sentences occurring and let the machine compose them into an overall statement. The characteristic will give the answer. (Figure 7)

Of course, the actual purpose of using "sufficient clues" is the correlation of hints and guesses in hopes of locating a document fitting a very narrow description.

Many of the decision-making procedures developed from symbolic logic principles can be readily adapted to the calculation of probability factors, instead of bare "yes" and "no" answers. In the present application to documentation, this is likewise the case. The algebraic equivalents through which the logic calculations are programmed for the computer permit the introduction of decimal fractions for probability factors, in lieu of 0 and 1 alone. Dependence must be set on the type of probability data obtained. If the relative probabilities were given in terms of the reverse relations, i.e. in terms of descriptors with respect to documents, rather than the other way round, then some appropriate transformation formula would be used.[x] Alternatively, without much extra effort, weighted multiple retrieval could be effected, i.e. each of several documents could be retrieved with an attached order of desirability. It would then be possible to have the machine judge through a numerical criterion. The operator may, on the other hand, prefer to use his own judgement as to the comparative suitability of his results.

Justification for many of the manipulations of probability factors arising in this field is beyond the scope of this paper and may be found in the Luce-Raiffa form of utility reference theory formulated by von Neumann and Morgenstern.

x) e.g., if $p(a:b_i)$ is the probability of a with respect to b_i, and $p(b_i:a)$, mutatis mutandis, then

$$p(b_i:a)=p(a:b_i)p(b_i)\Big/\sum_j p(a:b_j)p(b_j)$$

Some Aspects Of Machine Translation

In view of the parallel relation between the symbolic logic approach to both document encoding and information retrieval, it is easy to see the further parallel relation between these and machine translation.

The subject of encoding per se has not been discussed in the present paper. Suffice it to say that, for the construction of a flexible, economical and inclusive proposition-dictionary, the Semantic Code discussed by John L. Melton[5] appears to be exceptionally promising. Other approaches to this part of the information problem are also being considered.

Machine translation, (MT) especially translation from one natural language to another, has received a great deal of attention and many results have been reported to-date from various parts of the world. This subject deserves more than a few remarks at the end of the present discussion. As a matter of fact, machine translation should be treated in a report devoted to the subject of the utilization of essential scientific data from foreign countries. A brief outline of the contribution to MT of techniques based on symbolic logic analysis will be given here.

The main tasks in MT are:

1. Construction of a formal system (including word lists) for describing natural languages.

2. Definition and evolution of algorithms for transferring (translating) from one system to another.

3. Development of the principles necessary for programming and coding these algorithms into the machine.

Symbolic logic as a tool enters into all three problems, especially the first one. The third problem, being the one most closely related to the similar problem in information searching generally, responds directly to the symbolic logic treatment discussed above. In other words, the truth-matrix analysis, the binary characteristic, the group-theoretical relations among characteristics, the logic pattern matching technique are all applicable to the handling of information consisting of linguistic algorithms.

The second problem is often merged and sometimes confused with the first. This happens largely because of the difficulty of divorcing semantics from the compilation of appropriate word-lists. The development of the necessary algorithms is basically a logic problem. For the source-target language pair, the proper formula structure can be set up by symbolic logic. Here, of course, quantifiers play an increasingly important role. The search for semantic structure is, in fact, the prime effort in today's move away from the primitive word-by-word look-up idea of MT. Multiple meanings of words as well as grammatical structure of sentences cannot be taken into account without this search.

Some successful attempts have been made to attack the multiple word meanings by the statistics of word-frequencies. The grammatical structure of sentences is being widely investigated by means of morphology (word order and inflection) applied to both source and target languages.[12,13] It is expected that symbolic logic characteristics can here be used advantageously to differentiate in a very simple way between structures.

By matching the constraints in structure of the source language with those of the target language, meanings rather than word-messages could be translated and furthermore, the storage requirements of MT can be much reduced in this way. The size of a well-organized glossary is, in fact, another leading question so far left unanswered partly because of the uncertainty as to the eventual results of structure studies. In the Russian language, for instance, 86% of the running words are inflected, and the total number of entries required for one noun average from 6 to 10, and one verb may require 59 entries[12]. The split-glossary technique of Dr. Milos Pacak, of Georgetown University, based on morphological analysis, may provide a much needed short cut. The classes and subclasses proposed by Dr. Pacak can be controlled by logic considerations.

A system of "Semantic Semaphores" is in preparation by the author. This, it is hoped, will aid in the establishment of word relationships and the preparation of word patterns relating to the compilation of thesauri. These, of course, also c call for decisions based on familiarity with specific technical fields.

For input to the machine, a satisfactory approach may be that of W. R. Nugent[14] or one developed in the U. S. Patent Office[15]. Mr. Nugent uses pseudo-alphabets having mnemonic names, which can easily all be directed by one single language pattern, but permitting a symbol capacity of 5000.

Conclusion

Lest one become so deeply involved in theoretical considerations of a retrieval system as to lose sight of its practical requirements, he should keep in full view the logical economy with which human beings, such as a good research librarian or a good oral interpreter, work.

The dexterity of the specialized human mind in "homing" rapidly toward the search goal must remain the ultimate aim of the documentation researcher, until that aim is reached and can be surpassed by a machine of machines.

Too often, an information system is exhorted because it has a remarkably large file or an astute way of responding to one type of request. For that matter, a dove picking out someone's fortune from a large pile of envelopes is also remarkable, if somewhat one-sided. It is perhaps not so much what a human being looks for in searching a library, as what he discards along the way that should tip off those who wish to direct machines to do as well as human beings.

A student having studied up on a large variety of topics for a comprehensive examination, seems to expel from his mind, as soon as he sees the questions on the examination, all those topics which do not pertain to these questions. A trained librarian or a trained interpreter similarly cuts down with bold strokes the search area through a few significant decisions which rapidly brings him to the eventual choice. Many other people do this in other connections.

What the machine must be told is how to perform this "preliminary" narrowing-down process which enables it to undertake the search "proper" only after the most time-consuming blind-man's buff through the logic network has been carried out and the irrelevant paths have been discarded. Machine searching, similarly to machine translation, may be best divided into two stages: the rough search (or rough translation), and the fine search (or fine translation). No exact theoretical dividing line can be given.

In a human being, this narrowing-down process works by means of two-valued logic, and therefore it must work likewise in the machine. The question now arises: Can present-day machines or machines now being designed along well-worn lines and capable perhaps of expediting the "fine search", be expected to master the real executive-robot task of the "rough search"?

It is suggested that the answer is: No! The "rough search", the portion of the work that almost demands a chess-player's ubiquity and judgment coupled with far superior speed, is not likely to be performed adequately for some time to come. In machines, non-arithmetical design concepts with as yet unrealized searching skills will first have to be developed. It may be well to have a close look at one of the few truly progress-oriented machines now in existence, viz. the Searching Selector at Western Reserve University[16,5].

A system of "logic traps", as mentioned above, must be achieved which rapidly discards the classes of topics that are not wanted by answering yes or no to well-ordered questions. It will not do merely to make up for the lack of adequate binary-logic networks by larger and still larger files and by greater machine speeds.

Often, the aspiration to higher speed obscures the purpose of searching. Of course, the highest possible speeds are desired and, in fact, machine speed is in itself a result of the improvement of the logic search paths. Shortcomings in logic, however, cannot be covered up by fast "horsework", either in the machine or in the paper-and-pencil system which may serve as a program for the machine. It is as incongruous to expect good results from a hasty approach in this field, as it would be for some computer manufacturer to expect good results if he felt that not every one of the magnetic core elements in his computer's memory needs be built carefully just because there are so many of them.

As Dean Shera remarked, " The dim light of the electronic tube has led us ever faster along the wrong path. For it is not in speed, but in capabilities that the great promise of automation lies . . ."

One may note that transistors shed an even dimmer light.

<u>References</u>

1. R.G. Larkin and H.M.Semarne, "Symbolic Logic in Engineering", 14th Annual ACM Meeting, MIT, paper #77, September 3, 1959.

2. H.M. Semarne, "Symbolic Logic Decision Devices in Documentation" Proceedings of the International Conference for Standards on a Common Language for Machine Searching and Translation, 6-12 September 1959, Western Reserve University, Cleveland, Ohio.

3. D. B. Netherwood, "A Theory of Logical Groups", Thesis, Air Force Institute of Technology, March, 1957.

4. K. Dürr, "Lehrbuch der Logistik", Birkhäuser, Basel, 1954.

5. J.W. Perry and Allen Kent, "Tools for Machine Literature Searching" Vol. I of Library Science and Documentation. Interscience Publ., N. Y., 1958.

6. M.R. Hyslop, "Inventory of Methods and Devices for Analysis, Storage and Retrieval of Information", in "Documentation in Action" (ed. by J.H. Shera, A.Kent, and J.W. Perry).

7. H.R. Koller, E. Marden, H. Pfeffer, "The Haystaq System: Past, Present, and Future" in "The Design of New Systems-Mechanical Problems Reports, U.S. Patent Office, Washington, D.C., 1957.

8. M.E. Maron, J.L. Kuhns, L.C. Ray, "Probabilistic Indexing: A Statistical Approach to the Library Problem", paper #13 presented to the 14th National Meeting, ACM, MIT, September 1, 1959.

9. H.P. Luhn, "The Automatic Derivation of Information Retrieval Encodements from Machine Readable Texts" paper presented at International Conference for Standards on a Common Language for Machine Searching and Translation, Cleveland, Ohio, 6-12 September 1959.

10. M. Taube and Assoc., "The Mechanization of Data Retrieval", Documentation, Inc., 1959.

11. R.D. Luce, H. Raiffa, "Games and Decisions" John Wiley and Sons, Inc., N.Y.,N.Y., Chapter II, and Appendix I.

12. M. Pacak, "Morphology in Terms of Mechanical Translation" Georgetown University, paper presented at International Conference for Standards on a Common Language for Machine Searching and Translation, Cleveland, Ohio, 6-12 September 1959.

13. G. Herdan, "The Statistics of Structured Meaning", papers presented at the International Conference for Standards on a Common Language for Machine Searching and Translation, Cleveland, Ohio, 6-12 September 1959.

14. W.R. Nugent, "A Machine Language for Documentation and Information Retrieval", Itek Corp., paper #15 presented on 14th Annual ACM Meeting, MIT, 1-3 September 1959.

15. S.M. Newman, R.W. Swanson, K.Knowlton, "A Notation System for Transliterating Technical and Scientific Texts for Use in Data Processing Systems", Patent Office Research and Development Report #15, 22 May 1959.

16. Allen Kent, "Machine Literature Searching at the Center for Documentation and Communication Research", report presented at the International Conference for Standards on a Common Language for Machine Searching and Translation

FIGURE 1

TABLE OF TWO-PROPOSITIONAL FUNCTORS

Polish Prefix Notation	Tie-Symbol	Traditional Designation	Meaning	Characteristic Value
Na	$\bar{a}$	non	not a	10
Kab	a•b	et	a and b	0001
Aab	a⋀b	aut	a or else b	0110
Vab	avb	vel	a and/or b	0111
Pab	a↓b	Peirce's arrow	neither a nor b	1000
Eab	a≡b	par	a like b	1001
Cab	a⊃b	prae	if a, then b	1101
Sab	a/b	Sheffer's stroke	not both a and b	1110

FIGURE 2

TYPICAL SIMPLIFIED PROPOSITION KEY
(kept down to four propositions for each of 7 disjoint statements)

Physical Description

a = it is a liquid substance
b = it is a gaseous substance
c = it is a crystalline substance
d = it is a colored substance

Composition

e = it contains N
f = it contains O
g = it contains only C and H
h = it contains S or P

Melting or Boiling Point

i = it has a melting point (m.p.)
j = it has a boiling point (b.p.)
k = its m.p. is exactly 114 degrees
l = its b.p. is exactly 222 degrees

Solubility

m = its solubility division is W
n = aqueous solution acid to litmus
o = aqueous solution basic to litmus
p = aqueous solution has pH δ

Distillation

q = it distils
r = it loses water on distillation
s = it sublimes
t = it tars

Reactivity of Substance

u = distillate reacts with nitrous acid
v = it produces an acid
w = it reacts with phenylhydrazine HCl
x = it produces an amide

Reactivity of Derivative I

y = derivative I has m.p. of α degrees
z = derivative I has b.p. of β degrees
a = derivative I is water-soluble
o = derivative I is in solubility division γ

FIGURE 3

EXAMPLE OF TRUTH-MATRIX

$\{[(\overline{a\downarrow b})\vee c]\cdot(\overline{a/b}\vee\overline{b/c}\vee\overline{a/c})\}\vee(a\cdot b)\cdot c$

VKVNPabcNVVNSabNSbcNSacKKabc

a	0 0 0 0 1 1 1 1	
b	0 0 1 1 0 0 1 1	
c	0 1 0 1 0 1 0 1	
$a\downarrow b$	1 1 0 0 0 0 0 0	
$\overline{a\downarrow b}$	0 0 1 1 1 1 1 1	
$(\overline{a\downarrow b})\vee c$	0 1 1 1 1 1 1 1	X
a/b	1 1 1 1 1 1 0 0	
$\overline{a/b}$	0 0 0 0 0 0 1 1	Y
b/c	1 1 1 0 1 1 1 0	
$\overline{b/c}$	0 0 0 1 0 0 0 1	W
a/c	1 1 1 1 1 0 1 0	
$\overline{a/c}$	0 0 0 0 0 1 0 1	Z
yvwvz	0 0 0 1 0 1 1 1	
yvwvz	1 1 1 0 1 0 0 0	U
u·x	0 1 1 0 1 0 0 0	
a·b·c	0 0 0 0 0 0 0 1	
(u·x) v(a·b·c)	0 1 1 0 1 0 0 1	characteristic

FIGURE 5

QUANTIFYING OPERATORS

	Prefix Symbol	Name	Meaning	e.g.
1.	Πx	ALL-operator	for every x	ΠxCxy
2.	Σx	EXISTENCE-operator	for some x (at least one)	ΣxCxy
3.	Ox	UNIQUENESS-operator	for exactly one x	OxCxy
4.	Mx	MAXIMUM-operator	for at most one x (none or one)	MxCxy
5.	Jx	CLASS-operator	the class of all x's such that	JxCxy

FIGURE 4

TREATMENT OF REQUEST WITH WEIGHTING FACTORS

1) Proposition-Key:

a = rubber material for use in O-ring packing

b = the packing to be exposed to hydraulic fluid on the inside

c = the packing to be exposed to atmospheric oxygen on the outside

d = the hydraulic fluid is an organic phosphate ester

e = the packing to withstand 225°F hydraulic system temperature

f = the packing to withstand mechanical abrasion

2) Weighting Factors:

the relative importance of b may be .8

the relative importance of c may be .6

the relative importance of e may be .5

the relative importance of f may be .3

These weighting factors may be re-interpreted as probabilities that the given property alone fulfil to requirements of the material.

The entries to be searched will, likewise, be set up with their respective weighting factors, thus making possible a preference order among the disclosures.

3) Symbolic Statement:

KKKKKabcdef (with b=.8,c=.6,e=.5,f=.3)

4) Assertion Pattern:

1,.8,.6,1,.5,.3

FIGURE 6

RELATIONS AMONG CLUE-WORDS

Request:

Article both in English and concerning aircraft or spacecraft, written neither before 1937 nor after 1957; should deal with laboratory tests leading to conclusions on an adhesive, used to bond metal to one of these: rubber or plastic; the adhesive must not become brittle with age, must not absorb plasticizer from the rubber adherent, and have a peel strength of 20 lbs./in; it must have at least one of these properties: no appreciable solution in fuel and no absorption of solvent

Clue-Words:

English, aircraft, spacecraft, 1937, 1957, laboratory, adhesive, metal, rubber, plastic, brittle, plasticizer, peel-strength, fuel, solvent

FIGURE 6
(Continued)

Proposition-Key:

a = it is an article in English

b = it is an article concerning aircraft

c = it is an article concerning spacecraft

d = it is an article written before 1937

e = it is an article written after 1957

f = laboratory tests were run

g = conclusions on an adhesive were obtained

The Adhesive:

h = the adhesive is used to bond metal to rubber

i = the adhesive is used to bond metal to plastic

j = the adhesive may become brittle with age

k = the adhesive may absorb plasticizer from the rubber adhesive

l = the adhesive may have a peel-strength of 20 lbs./in.

m = the adhesive may have appreciable solution in fuel

n = the adhesive may have absorption of solvent

Symbolic Statement:

KKaVbcPdeCfg
KAhiKKKNjNklSmn

FIGURE 7

A "SUFFICIENT CLUES" PROBLEM AND SOLUTION

x,y,z = documents written by authors x,y,z, respectively
α,β,γ,a,b,c = documents α,β,γ,a,b,c, respectively
p,q,r = the topic of the document is one, two, three, respectively
m = the topic of the document is mathematical

From statements 1. and 2. of the Problem, one gets 6 sentences of this form: AAxyz, all connected by conjunctions, and 18 sentences of this form: CCαxKCβAyzCC$_{βγ}$Cγz, 9 for α,β,γ and 9 for a,b,c; all connected by conjunctions

From statements 3.,4.,5: Ccp,Cyq,CbNm,
From statement 6, one gets 3 statements of the form: CCCyαar
From statement 7: KKCCamayCCbmbyCCcmcy
From statement 8: CαNz

The total characteristic of 2^{13} bits simplifies to an assertion pattern of 13 bits which indicates the presence of the relationship VNαx or Cαx, i.e. $α \supset x$