

# **HIERARCHIC HEURISTICS:\***

Their relevance-to economic pattern-  
recognition and high-speed  
data-processing

Leonard Ornstein, Ph. D.

Dr. Ornstein is Professor of Pathology (Biology) and Director of the Cell Research Laboratory of the Graduate School of Biological Sciences of the Mount Sinai School of Medicine of the City University of New York.

\*This paper was written in 1969 for the November issue of *Proceedings of the IEEE*, but was never published.

## ABSTRACT

Economic and intellectual motives exist for trying to use computers for the resolution of questions that typically require resort to vast stores of empirical experience in the search for their answers (e.g., questions which may require rapid random access to reference files containing more than  $10^8$  bits of data). Such questions can be recast as pattern-recognition problems (43, 44, 25). They range from the routine and intellectually trivial, such as word recognition in ordinary speech, to tasks which can tax the intellectual capacity of the trained professional, such as the working out of a medical diagnosis. As of this time, no economic or adequately quick solutions to such problems, even using late third-generation computers, have been proposed. I will attempt to establish some confidence in the proposition that such pattern-recognition problems might be solved economically by the application of hierarchic principles. Important problems in telecommunications coding also will be shown to find novel solutions within the hierarchically structured solution to the pattern-recognition problem. The general pertinence of large and fast hierarchically structured computer memories to real-time solutions will be discussed and a new design for such a memory will be briefly described (49a).

## INTRODUCTION

We deal with hierarchical structure daily. Our governments, institutions, businesses and families are hierarchically organized and we often arrange the explanation or teaching of simply structured subjects in hierarchic sequence with the intuitive understanding that kinds of heuristic efficiency are achieved in this way. Yet logical and algebraic properties of hierarchies are not discussed in our general or scientific education. The properties of hierarchies have classically and mainly been of concern to, and have received attention from biologists (e.g., morphologists, taxonomists and occasional medical diagnosticians). More recently, those concerned with the design, development and management of inventories, directories, indices, libraries, bureaucracies, computer languages, sorting techniques and file structures [e.g., see (33) and (37)] have made considerable use of hierarchical techniques. Using insights drawn from the continuing concern of biologists with the nature of taxonomic classifications [e.g., (39a, 50)], I will try to demonstrate that near optimal solutions for a large class of coding problems (for automatic classification of all kinds of objects, events and concepts) can be formulated by incorporating the valuable properties of hierarchies into appropriate models. The classification of diseases plays a

fundamental role in the diagnosis of ailments and prescription for therapy, the two activities central to the delivery of health services. The relevance of these models to the development of efficient software and hardware for automated medical diagnosis capable of helping in the delivery of higher quality and cheaper health services is of no minor practical concern in this discussion. Based on considerations which will be detailed in the body of this paper, it can be estimated that the cost of the necessary research and development will be monumentally steep if there is to be confidence of achieving real success. In view of the current declining levels of investment in health research and development, the economic motivation to underwrite a substantial portion of such research and development instead may be stimulated by the potentials of hierarchical models in communications engineering. For this reason, the main focus in this paper will not be on models designed specifically for application to biomedical tasks, but on models for real-time coding and decoding procedures applicable to telephone communications, voice-operated typewriters and voice input to computers.

In communications engineering, the job of relieving the growing overload of telecommunication channels increases in priority as each day passes. More and higher capacity channels (31a), on the one hand, and better utilization of channel capacity, on the other, are the two routes to relief. By 1968, about one half of all Bell System long-line telephone circuits were being used for service other than telephone messages (e.g., T.V., facsimile, digital data, etc.). It has been estimated that by 1970, the required capacity for wide-band, non-telephone use will have doubled that of 1966, while the capacity needed for telephone service will have increased by about 20% (52). To begin, in part, to meet such needs, A. T. & T. has proposed a \$170,000,000 coaxial cable with a capacity of about 105 analogue telephone channels between St. Louis and Los Angeles<sup>1</sup>. This is intended to meet the problem by the first route. Let us examine the second route, better channel capacity utilization.

In 1949, in his classic paper on the Theory of Communications (59), Shannon defined an intuitively satisfying and explicit quantitative measure of information. As a corollary to that definition, the average rate of generation of information by a message source is  $-(\sum p_i \log_2 p_i) / (\sum p_i t_i)$  bits per second, where  $p_i$  is the probability of occurrence, and  $t_i$ , the duration of the  $i$ th kind of statistically independent segment of a message produced by the source. He proved that the information transmission capacity of a communication channel is  $W \log_2(1 + S/N)$

---

<sup>1</sup> Washington Newsletter, *Electronics* **41**, Oct. 14, 1968, p. 86.

bits per second, where  $W$  is the bandwidth of the channel in hz., and  $S/N$  is the power ratio of signal to noise in the channel. He also proved that, given some memory device (properties unspecified), defined segments of messages from an information source can be coded so that the channel capacity required to transmit such messages can be made to approach, as a limit, the rate of generation of information by that source. In general, the greater the average length of the coded segments, the more nearly will the required statistical independence criterion be met, and therefore, the closer will the limit be approached (but the delay incurred in the coding and decoding procedures will necessarily increase). The rate of generation of information by a message source divided by the actual capacity used to transmit messages is Shannon's measure of efficiency of channel capacity utilization. (He also proved that the messages from a source can, in principle, be transmitted over a channel with unity efficiency with as small an error as desired, again at the cost of increasing coding and decoding delays. Although this aspect of his theory has received major attention in the communications' literature [e.g., see (51)], it is not at the main focus of this discussion.) Discussions of Shannon's work usually neglect explicit reference to the fact that some pattern-recognition procedure is a *sine qua non* for defining both the boundaries, and perhaps more importantly, the kinds of segments in messages from an information source prior to any coding procedure that might satisfy his coding theorems. Thus, such efficient coding algorithms as Huffman's (26) assume already defined and recognized kinds of segments. This may in part explain why the most important unanswered question in communication engineering has remained, "How can the promise of Shannon's 'existence theorem' on matching sources to channels be realized operationally and in real-time?" I will develop arguments that hierarchical taxonomic pattern-recognition procedures coupled with hierarchically organized hardware for their efficient execution, provide an operational answer for typical information sources.

In the introduction to the same work, Shannon stated quite emphatically that the meaning of a message (or of its parts?) is irrelevant to the solution of the engineering problem of efficient and economic channel utilization. An important way in which meaning becomes relevant to the efficient matching of message sources to a communication channel is also uncovered by an examination of hierarchical taxonomic procedures.

The hierarchical pattern-recognition procedures of our models should permit the transmission of quality band-limited speech over communication channels with as little as 50 bits per second capacity. This capacity must be

compared with the present requirement for 20,000 bit per second voice channels (bandwidth approximately 2,500 hz., S/N, 256/1; i.e., 24 db.) for typical telephone transmission; and compares very favorably with the most promising "vocoder" channels now receiving the attention of communication engineers [e.g., (16,28,42)]. The economic potential of our models, and therefore a measure of the incentive for their realization in software and hardware, may be appreciated by a brief examination of the possible savings in long-distance common-carrier communications:

If the telephone messages to be transmitted over the proposed \$170,000,000 St. Louis to Los Angeles cable could be coded down to about 50 bits per second at one end, and then were multiplexed and transmitted in a time-sharing mode, and finally were separated and decoded at the other end, then the capacity for telephone use would be increased by about 200 times, thus almost doubling the available wide-band capacity of this cable.<sup>2</sup> Translated into the economics of research and development, the investment of as much as \$100,000,000 for the realization of such a model would be more than justified, not to mention that it would permit more efficient utilization of existing long-lines and would reduce the need for future capital investment in other long-lines and communication satellites by perhaps billions of dollars. (Though perhaps not so economically obvious as this example, even larger economic gains may be achieved outside of the areas which classically have been identified as parts of communication engineering, such as the aforementioned biomedical applications, including automated medical diagnosis.)

In this paper, I will also attempt to show that the nature of the computations required for this varied and extensive class of applications supports a change in emphasis in the design of general purpose computers, from sequential machines to machines with very large-scale parallelism (26a), to meet what are likely to be the needs of the major data processing markets of the future.

Hierarchical structure permits economies in the design of both software and hardware:

Although hierarchical structuring of software has been fairly widely used as an *ad hoc* device for reducing the number of logical processing steps necessary to complete such operations as a search by comparisons [e.g.,

---

<sup>2</sup>. This calculation assumes that each 20,000 bit/sec. analogue voice grade channel to be used for coded voice communication has been converted, for example, by existing adaptive digital equalization techniques (30), to a 9,600 bit/sec. digital channel.

the well known binary logarithmic sort (43)], the possibility that some hierarchical software structures will lead to the class of information-theoretically-efficient coding and decoding procedures referred to above, has not been generally appreciated. Aspects of the efficiencies derivable from hierarchical software will be examined in **Sections I, II, III, VI, VII and IX**.

Coincidentally, hierarchical structuring of hardware, as well as other constraints on hardware design, permit the execution of the hierarchically structured software procedures (and many other non-hierarchical procedures) in a minimum amount of time and with reduced hardware costs. Aspects of such hierarchical structuring of hardware will be discussed in **Sections III, IV, V, VI, VII, VIII and X**.

I believe parts of early sections will represent the first explicit exploration of a few heuristically familiar processes. This will entail what may seem to some readers to be an inordinate amount of attention to rather prosaic details. As a result, since “familiarity breeds contempt”, it would be relatively easy to underrate the importance of, and to overlook what I believe are the powerful consequences of such an exploration. I hope the reader will bear with me when I seem didactic.

This discussion will draw heavily from a previous work (49) in which many details and general implications are covered quite explicitly in a more biomedically oriented model.

## **I Taxonomic Keys**

In part, biological taxonomic classification has been canonized in the structure of memory files known as taxonomic keys [e.g., see (68)].<sup>3</sup> Ideally, in such a key, a minimal set of typical (diagnostic) attributes (that is, attributes which are shared in common by all the organisms which are members of a branch of the taxonomic tree but which are not present among the members of other branches arising from the nearest common branch point), are listed under the name of the appropriate branch. The branches, in turn, are subdivided into sub-branches which contain members more similar to one another within a sub-branch than to members of other sub-branches, and the diagnostic attributes of each sub-branch are again listed under the appropriate name of the sub-branch. This process is

---

<sup>3</sup> A good deal of “art” enters into the process of constructing taxonomic keys in the sifting out a small subset of diagnostic or typical attributes or descriptors from a set of an enormous number of attributes. This will be discussed in **Sections VI and X**.

repeated a number of times out to the branch “tips” (for example, in biological classification, out to Species).

We will begin by examining an idealized but unsophisticated taxonomic key and some of the operations performed in its application to routine recognition (or identification or naming) tasks . (Later we will examine the problem of “typicality”.) Such tasks are not necessarily trivial undertakings. For example, when we have properly named an “unknown”, we usually can turn to a memory file other than the taxonomic key [e.g., an encyclopedia, a therapeutic index (36), etc.] . Using the discovered name (or “diagnosis”) as an address or index term, we can gain ready access to such a file and retrieve considerable and often useful (though perhaps not “typical”) data on the attributes of the named sample, (for example, in the case of a pathogenic organism, data on its susceptibility to various drugs and, therefore, one means for “prescription for therapy”). This, in fact, is one of the main functions of classification, naming and coding .

Let us suppose, for simplicity (39a,67), that there are 9 main branches and 10 sub-branches at each branch point of 6 succeeding levels of branching of the taxonomic tree, (e.g., seven such taxonomic levels would be: Kingdom, Phylum, Class, Order, Family, Genus, and Species). Let us number the first 9 main branches of the tree from **1** to **9**. The 10 sub-branches of branch **1** are numbered from **10** to **19** ; the 10 branchlets of sub-branch **10**, are numbered from **100** to **109** and so on. We assign one page of the key to each list of diagnostic (typical) attributes and we number each page with the number of the appropriate sub-branch . We finally place all of these pages in numerical order **1** to **9, 999, 999**.

Given an unidentified organism, we sequentially examine each of the pages, **1** to **9**, comparing the organism’s attributes to those listed to discover to which Kingdom of organisms it is most similar or which it best “matches”. Let us suppose that the unidentified organism is, in fact, a black-widow spider. As a result of the examination of pages **1** through **9**, we find a best match with the sub-set of attributes listed on the page which represents the Kingdom, *Animalia*. Let us suppose that is page **2**. We then skip to pages **20** through **29**, which contain the descriptions of all the animal Phyla to discover to which it belongs. If, for example, page **28** lists the attributes of the Phylum of all animals with more than four jointed-legs, i.e., the *Arthropoda*, we will find our best match there. We now skip to pages **280** through **289** which will contain all the Classes of the Phylum *Arthropoda*. Included will be the Class of the 8-legged *Arthropoda*, namely, the Class, *Arachnida*, which will provide the best match at this level. The process is iterated, requiring the examination of the contents of 9 pages at

the first level and 10 pages at each of 6 succeeding taxonomic levels and therefore only 69 pages instead of about 10<sup>7</sup> Pages to identify the Species of organism.<sup>4</sup>

Note that the first (left hand) digit of a page number (code number) designates to which Kingdom the organism belongs. All animals have code numbers beginning with **2**. We may have enough information about our unidentified organism to completely “resolve” its species, or with less information or less time, only identify the order or family. Nonetheless in either case, after the first successful comparison has been made at the Kingdom level, we will have already “read” the first digit as a **2**. *Homo sapiens* (a Species) would be represented by a 7 digit number beginning with a **2**, *Mammalia* (a Class) by a 3 digit number beginning with a **2**, and *Arthropoda* (a Phylum) by the two digit number, **28**.

## **II. Significant and Nonsignificant Codes, Meaning and “Dictionary” Efficiency**

A code with code words having the property such that at least two symbols in a word each signify some different members of a characteristic set of properties or attributes of the object, event or concept encoded is defined as a significant code (14). Conversely, if a code is constructed so that there is no fairly direct relationship between the individual symbols and the properties of the object they represent, it is called a nonsignificant code. Clearly, the taxonomic code outlined above (the page numbering system) is, by definition, a significant code. The alphabets of all languages constitute nonsignificant codes unless the phonetic sound of the word, rather than its “dictionary meaning”, is considered to be the set of properties coded for by its spelling. Then, for example, the phonetic portion of an English dictionary is a code book for a somewhat ambiguous <sup>5</sup> but nonetheless significant

---

<sup>4</sup> In a recent and perceptive review of some aspects of such questions as how much computation, how much memory, how much time, what class of machines and what degree of complexity are necessary for information retrieval and inductive inference by machine, Minsky and Papert have stated, under the chapter heading, Gloomy Prospects for Best Matching Algorithms, “..when we turn to the best match problem...we conjecture that...for large data sets with long word lengths there are no Practical alternatives to large searches that inspect large parts of memory.” [Emphasis that of Minsky and Papert (43).] They appear to have overlooked the significance of hierarchical taxonomic searches, for they cannot consider 69 parts out of 10<sup>7</sup> to constitute “large parts of memory”. Their gloomy conclusion clearly does not apply to the many interesting best match problems where the objects to be classified are distributed in tight or only slightly “fuzzy” clusters in an n-dimensional space, but almost certainly does apply to the case where the objects are randomly distributed in such a space. It is the large “gaps” between “natural” clusters (49) typical of populations amenable to hierarchical taxonomic classification which considerably simplify the best match problem (see pages 33 - 34 and 35 - 36).



<sup>5</sup> i.e., pronunciation rules in English have numerous exceptions; however, see (23).

code. However, most words of a language are not significantly coded with respect to their dictionary meanings. For example, there is no relationship between the individual letters, or combinations of letters, in the word, “house”, and any of the attributes of a house. On the other hand, many word-roots can be considered to be significant code symbols which are signs for attributes of the objects they designate; for example, *Arthropod*,--- jointed-foot. Codes based exclusively on statistical properties of a data set [e.g., Huffman Codes (26), etc.], and such codes as exhibit an essentially “random” relationship between the symbols and the object properties (e.g., accession numbering), are nonsignificant codes.

Two properties of some significant codes are especially interesting because of potential value in automatic data processing in telecommunications:

1) Those which have a hierarchic structure provide substantial economies in both coding and decoding (or often equivalently, as indicated in **Section I**, search and retrieval steps).

2) Shortened “messages” made up of taxonomic code “words” in which suffixes have been dropped, can be both transmitted over communication channels with less capacity than required for the unshortened messages and can be decoded with the same dictionary (code book or key) as used for “full-length words”, with reduced but still meaningful “semantic resolution”.

In contrast, truncated code words of a nonsignificant code (e.g., a Huffman code) yield meaningless messages with the nonsignificant decoding procedure (49). For nonsignificant codes, separate code books could be supplied for decoding messages subject to each different degree of truncation, but only at the price of greatly increasing the amount of memory needed. The insertion of check symbols (51) into significant taxonomic code words at regular intervals along the word, with each succeeding check symbol serving as a check on the last contiguous sequence and on all preceding sequences in the word, provide considerable resistance to the effects of reduction of channel capacity [for example, those due to decrease in the signal to noise ratio of the channel (63)]. This results because, as errors accumulate, the confidence with which a received decoded word can be correctly interpreted will increase as one moves back from the end of the word towards its first symbol. Thus a taxonomic significant code, with properly incorporated check symbols, maximizes transmitted meaning over channels of uncontrolled capacity [such as a typical analogue telephone channel] (49). These aspects of the relationship of meaning to efficient channel capacity utilization have been discovered by an examination of some details of the kind of memory device which Shannon left unspecified. Both of these properties will be examined again further on in the text.

### III Taxonomic Keys and Computer Memory Organization

We will now examine some of the operations that might be employed to classify an organism by means of a taxonomic key to illustrate further some sources of hierarchical efficiency. Let us again suppose that the unknown organism to be identified is a black-widow spider. Having found a closest match with the attributes on page **2** of the key, a person performing a search of the key would skip pages **10** through **19** before searching further. He also knows, from the last step, even before finding the next match somewhere among pages **20** through **29**, that only pages having a number beginning with a **2** will be pertinent to this search and that he will be able to skip at least pages **30** through **199**, and so on. At each taxonomic level, he will automatically become apprised of an increasing number of the pages of the key which can safely be skipped.

As noted previously, the key is a memory file. If all the information in our hypothetical taxonomic key were contained in a random-access memory, the large number of pages which can be "skipped" would permit substantial increase in the speed with which a specimen might be classified. However, we might require something like  $2 \times 10^9$  bits ( $10^7$  pages, 200 data bits/page<sup>6</sup>) of, for example, magnetic core memory, for this particular key, in order to be free of page-skipping speed restrictions. Even if we settled for a relatively cheap magnetic core memory, at a cost of a few cents per bit, the memory cost for the core stacks, neglecting the costs of auxiliary control hardware, would be many millions of dollars! If we choose, instead, strictly serial memory (for example, 800 bits per inch, 9-track magnetic tape, transported and read at 180 inches per second) the memory cost will only be about \$600 for about 48,000 feet of tape required, but the access time to, for example, the last 200-bit entry will be on the order of  $10^7$  times longer than in the case of random-access core memory which requires the preliminary reading of only 68 other pages (e.g., about 1 hour as compared to about 100  $\mu$ sec.) because no page skipping would be possible.

It is now generally appreciated that the kind of memory organization needed for efficient execution of this type of hierarchic search is, often itself, hierarchic. For example, if the  $10^7$  page taxonomic key were, in fact, distributed sequentially in 100 separate 480-foot reels of 1/2-inch wide magnetic tape, numbered from 0 to 99, the pertinent pages through Family (the first  $10^5$  pages, all contained on the first reel) would be accessed in about 1/2 minute. In the case of the spider, even before the first 3 inches of

---

<sup>6</sup> The minimum requirement would be over 22 bits of information per page.

tape had been read, we would have discovered that the specimen was an animal and that we would therefore have to go directly to the third reel, No. 2, which would contain pages **200,000** to **299,999**, when the search of the first was complete, and after that, to one of the set of reels, Nos. 20 to 29. Precisely which, would also have been determined after reading the first **29** pages on the first reel (within the first 10 inches). Thus, the total tape scanning time would be reduced to something less than 2 minutes by subdivision of the tape into 100 reels. The time to read reel numbers, skip the non-pertinent reels and manually select and mount the pertinent reels on the tape transport would take about a minute.

**Table 1** shows the search speeds and some costs associated with various kinds and divisions of storage media, indicating that the preferred currently available mass medium for this application would be something like RCA magnetic cards (19), the magnetic strips of an IBM data cell (62), or such recently announced photo-optical mass memories as the FM 390 (32).

#### **IV “Almost Parallel” Search At Different Levels of a Hierarchical File**

Parallel processing can help provide still faster search. If our key were stored on 1000 magnetic cards, and if we have four card-selector-readers under independent parallel control, then as soon as the first card has been in part read, the second, third and fourth pertinent cards would be identified, selected and advanced to addresses fairly close to those required to complete the search, using the results obtained from the early part of the search of the first card. For example, processing of the first card will sequentially provide 4 digits, the first would identify the second pertinent card; the first and second digits would identify the third pertinent card; and the first three digits would identify the fourth pertinent card. The total search time, therefore, would be reduced to something near 1 second, i.e., the access and read-time for one card.

In a serial memory, we pass by each entry at the same speed, that is, we sequentially scan every “word” of every “line” of every “page” of the key. In an ideal hierarchically organized serial memory, we would have different sized blocks of data (serially organized) at each level, and we might have independent scanning modes for skipping and accessing the various size blocks (e.g., symbols, words, lines, paragraphs, pages, chapters, books, shelves, library stacks, library floors, libraries, etc.). Efficient access to the contents of a large hierarchical memory file of sequentially searched and

**Table 1****Combined Memory Search & Reading Times and Storage Medium Costs to Retrieve****One Out of  $10^7$  Branch Tip (200 bit) Pages of a Decimal Taxonomic Tree****Using Various Types of Mass-Memories (1969)**

Medium Type, $2 \times 10^9$ bits	Approx. cost of <u>Medium</u> \$	Approx. Machine <u>Search Time</u> to Retrieve 1 page	Number of Human Operator Interventions	Time per Human Intervention	Approx. Total <u>Search</u> <u>Time</u>
Random-access mag. cores, 1.5 $\mu$ sec. read time (200 bit words)	$6 \times 10^7$	100 $\mu$ sec.	0	0	100 $\mu$ sec.
Serial search of one 48,000- foot reel of 1/2" 9-track 800 bpi tape	600	1/2 - 1 Hr. (tape transport speed, 180 ips)	0	0	1/2 - 1 Hr.
Serial search (with skipping of reels) of 100 480-foot reels of 1/2" tape	600	1 - 2 min. (tape transport speed, 180 ips)	3	40 sec.	4 - 5 min.
Serial search (with skipping of reels) of 1,000 48-foot reels of 1/2" tape	600	6 - 12 sec. (tape transport speed, 180 ips)	4	45 sec.	3.2 min.
30 Disk-packs, $6 \times 10^7$ bits per pack (with skipping of packs)	15,000	2 sec.	2	30 sec.	1.1 min.
1,000 mag. cards, $2 \times 10^6$ bits per card (with skipping of cards)	2,000	4 sec.	0	0	4 sec.
60 FM 390 photo- data cards, $3 \times 10^7$ bits per card (with skipping of cards)	15	3 sec.	0	0	3 sec.

---

Note : In 1997 the entries in **Table 1** look like this:

Medium Type, 2 x 10 <sup>9</sup> bits	Approx. cost of <u>Medium</u> \$	Approx. Machine <u>Search Time</u> to Retrieve 1 page	Number of Human Operator Interventions	Time per Human Intervention	Approx. Total <u>Search</u> <u>Time</u>
60 ns DRAM ( 12 Mbytes)	50	1.5 µsec.	0	0	0
12 MBytes of Hard Disk	0.10	10 ms.	0	0	0

---

ordered blocks of data can be facilitated by a number of hardware features. The memory at each taxonomic level of a taxonomic tree might be allocated to a different device which can be searched and read independently and in parallel with the others. The block-size utilized at each taxonomic level should be equal to the total size of the data file at the next level closest to the tree trunk. (In our previous example, there would be 10 blocks of data at each memory level other than the first.) The total contents of a block should, if possible, appear to be simultaneously accessible (as is typically the case for whole “words” in a random-access memory) only requiring identification of the address of the block (for example, the first page number in that block).<sup>7</sup>

We will now examine how such a memory structure might be utilized for our identification task:

The time it takes to read the contents of 10 pages, compare each to the “description” of the unidentified sample, measure a set of 10 “similarities” and then determine which of the similarities is greatest, sets a limit to the maximum useful speed of transfer of a block of data from one level of memory to another. Let us suppose that a block of 1000 pages of the key can be located and transferred from some relatively cheap mass-memory [e.g., a Photostore (29)] to disk, and simultaneously, 100 pages can be transferred from disk to core and 10 pages from core to a fast semiconductor “cache” memory (21,9) each within the limiting time just defined. Our strategy might be as follows: We store the first 9 pages of the key in the “cache” and reserve one “empty” page in the “cache”; the next 90 pages go into core, with 10 reserved empty pages; the next 900 pages on

---

<sup>7</sup>. The requirement merely to read a block address and transfer the entire block permits potentially faster, simpler and cheaper hardware configuration than for usual pseudo “random-access” memories (8a).

disk with 100 reserved empty pages, etc.. When the first 9 pages have been read and the Kingdom (for example, *2, Animal*) has been identified, pages **1** to **9** are replaced with pages **20** to **29** (utilizing the empty page). This buffering into fast memory should be arranged to occur in a time that is short compared to the processing time. Processing of pages **20** to **29** commences immediately. With essentially simultaneous, operation of the control devices for each level of memory inparallel, we also transfer pages **200** to **299** from disk to core, replacing pages **10** to **99** and filling the empty spaces; transfer pages **2,000** to **2,999** to disk, replacing pages **100** to **999**; etc.. The entire process is iterated after each higher level of the tree has been searched.

With this kind of hierarchic organization of memory, with simultaneous parallel access and transfer of blocks of data, the problem could be executed in very nearly the same amount of time as if the entire key had been stored in expensive fast random-access memory, but with a much more modest investment.<sup>8</sup>

## **V A Model for a Hierarchical Memory for Fast Random-Access to Large Blocks of Data (49a)**

Consider the schematic, in **FIGURE 1A**, of a “relay” tree or “nerve net” similar, in principle, to memory address trees. Let us suppose that the array is a miniaturized printed circuit. The conductors of the tree are tin and the control lines, labelled with **0**'s and **1**'s, are lead. The entire assembly is held below the critical temperature of tin. Each “relay” element is then a crossed-film-cryotron<sup>9</sup> (13). When the appropriate current flows in a lead control line generating a magnetic field greater than the critical field of tin, all the underlying tin lines are switched from superconducting to normal states, “closing” those “gates”. For simplicity, we will assume that the input ends of the tin wires are either permanently connected to the input or not, to store a **1** or **0** respectively. This, then, is a random-access read-only memory.<sup>10</sup>

---

<sup>8</sup>. By doubling the memory at each level, pages **1** to **9**, **10** to **99**, **100** to **999**, etc. could remain in place so that recycling time for search for the next organism could be negligible, (otherwise it would take about one complete search time to restore the memory to the starting condition).

<sup>9</sup>. In-line controls (rather than crossed-films) in a low-inductance pattern with shield planes (8,13) would probably be used, however the illustration would then be excessively complex.

<sup>10</sup>. With modification it could also be designed with erasing and writing capabilities.

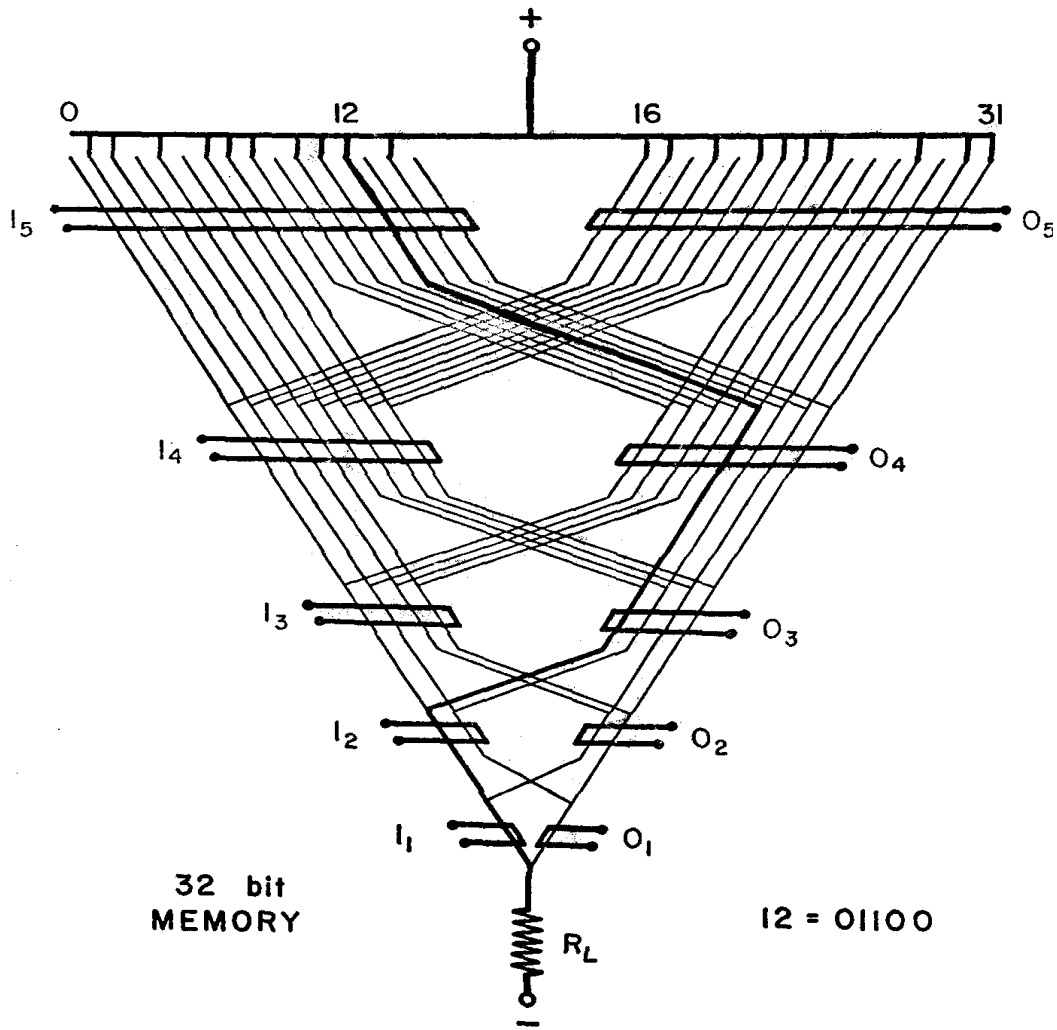


FIGURE 1A

**FIGURE 1A.** A 32 bit superconducting hierarchical "Address-tree Memory". Address lines, labelled  $0_1, 0_2, 0_3, 0_4, 0_5$  and  $1_1, 1_2, 1_3, 1_4, 1_5$  are made of lead. The tree conductors are tin. When only address lines  $0_5, 1_4, 1_3, 0_2, 0_1$  draw sufficient current to switch the lines they cross to the normal resistive state, the only remaining zero resistance path through the tree will be that leading from memory address,  $1_2$ , to the load resistance,  $R_L$ . If the load draws current, a 1 is read from the accessed address; otherwise a 0.

If there are  $M$  branches at each branch point and  $L$  levels of branches, (and therefore switches) ( $M = 2$  and  $L = 5$  in the illustrated case) then any one of the  $ML$  stored bits can be addressed and read out across the load,  $R_L$ , by activating  $L$  out of  $ML$  appropriate control lines, one at each level. For example, the bit stored at address  $1_2$  (a one) is read out as a voltage across  $R_L$  by passing current through control lines  $0_5, 1_4, 1_3, 0_2$  and  $0_1$ , at levels 5, 4, 3, 2, and 1 respectively, effectively blocking the conductivity of

the included bundles and leaving only one zero resistance path from the tree top to RL. (Note that the control line code, **01100**, is **12** in binary notation.)

Now consider a binary memory tree of this kind where  $L=31$ . (The capacity of such a tree,  $M^L = 2 \times 10^9$  bits. Such a tree could be constructed to occupy no more than a few cubic feet of space.) Next consider a tree in which the 14 highest levels are identical to this latter tree, but in which the lower 17 levels are missing and each lower end of a tin “branch” feeds its own load,  $R_i$  (A simplified version of such a tree is illustrated in **FIGURE IB**.) This truncated tree top could, with a 14-bit binary address, transfer the contents of any one of about  $10^4$  blocks of 1000 pages (200 bits per page) of the key simultaneously to its output loads. In effect, a large block of data (half the memory, when  $M = 2$ ) is transferred from the top of the tree to the next level; a selected subset of this set (again a fraction =  $1/M$ ) is transferred to the next level, and so on, all simultaneously, so that this single memory can be considered as equivalent to the hierarchical set of memories sketched out in the preceding section and feeding the fast semiconductor “cache”. The use of address-tree branches for the sense-lines themselves, rather than for control lines, appears to be unique in large-scale <sup>11</sup> digital memory design. The truncated design also easily gives simultaneous access to words (blocks) of virtually any size. Even if the cryotron switches were relatively slow (e.g.,  $10^{-4}$  sec.) such a device could vastly increase the apparent high-speed random-access capability of a present generation general purpose computer, if used as described in the previous section.

Another type of memory, which is being developed for all-electronic telephone switching systems, and which permits fast random-access to large blocks of data, is a laser-scanned holographic memory (2,64).

So-called content-addressable or associative memories, which are constructed with independent logic elements for each memory location (or set of memory locations) permit direct access to a memory location in one step (e.g., by using the data describing an organism as the address to the Species page of the key) (35). The cost per bit for such memories is much greater than for a bit of magnetic core because of the added costs of the logic circuitry associated with each memory cell. Therefore, until some entirely new form of associative memory is proposed, the cost of building very large files in associative assemblies will not be able to compete with hierarchically structured memories or even with direct-access mass

---

<sup>11</sup> Classical diode decoders, which operate in this way, are not useful for large-scale devices because of the very large dissipative loads such arrays would represent. Large cryogenic trees, by comparison, are essentially zero-loss devices.

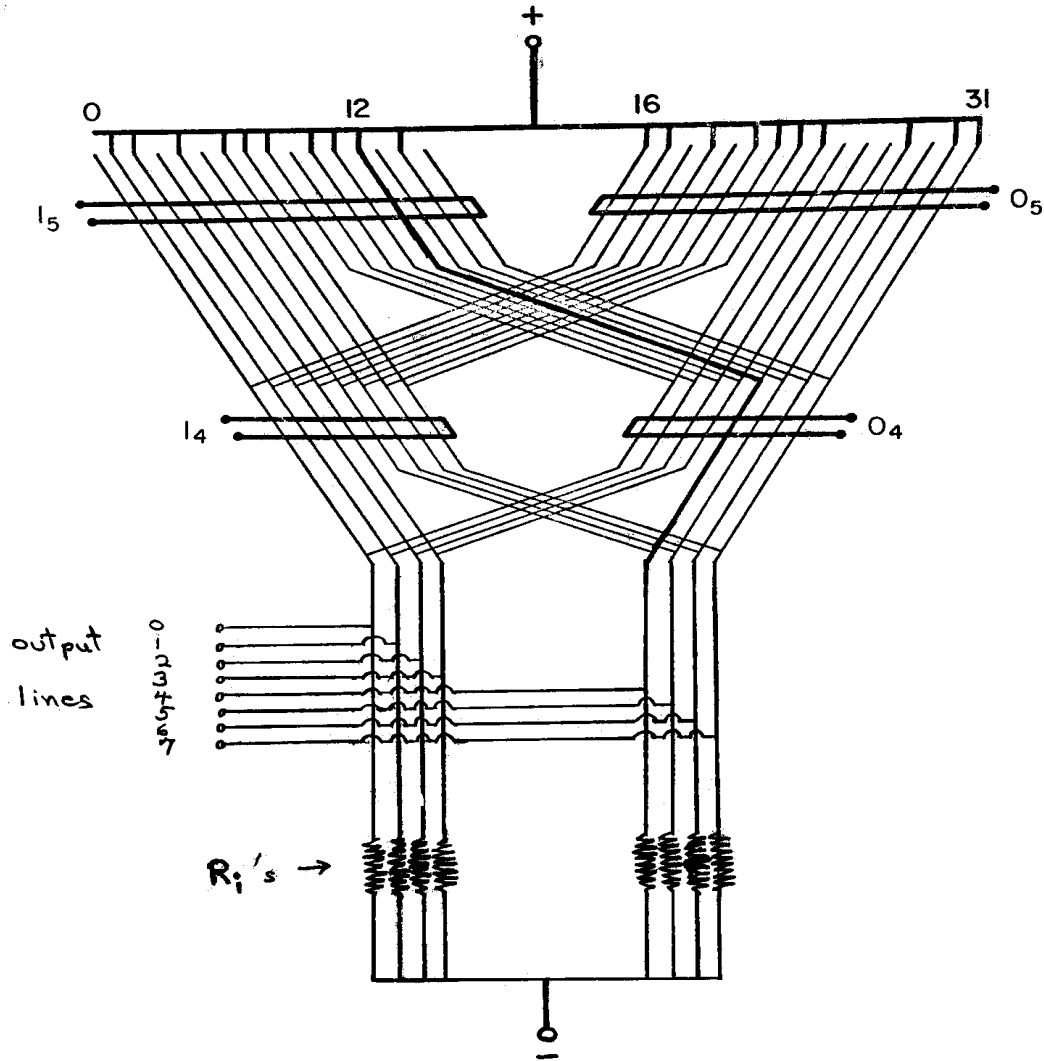


FIGURE 1B

**FIGURE 1B.** "Truncated Address-tree Memory". With switches **05** open (and **15** closed) and **14** open (and **04** closed), all **ones** stored in the block of addresses, **8 - 15** (all of which addresses, when represented in 5 bit binary notation begin with **01...**) will appear as positive voltages when measured between the appropriate out-put lines, **0 - 7**, and the negative terminal (i.e., voltages will appear on out-put lines **0, 2, 3, 4,** and **6** representing the **ones** stored in addresses **8, 10, 11, 12** and **14**).

magnetic core memories (57,8a).

The sections which follow will show that, for optimization of search-speed, degree of parallelism and cost, different numbers of branching points (and therefore blocks) are required at each level, depending upon the speed of access to, transfer of and processing of the contents of each kind of block.

## VI Optimum Hierarchic Efficiencies

Suppose we draw samples at random from a very large <sup>12</sup> population of lists or vectors (representing objects, events, or concepts) composed of  $N$  distinct and equally frequent classes which could be defined such that individual members of a class (based on some “comparison” of, or measure of, or operation on, etc., a set of  $X$  “corresponding attributes”<sup>13</sup>) are “more similar” or “closer” to one another than to members of other classes. There are 2 cases of interest:

### A. “Ordering a Familiar Universe”

In this case, we have prior knowledge (definitions) of the  $X$  corresponding attributes for each of the  $N$  known sample classes.

### B. “Learning Without a Teacher”

Here, we have definitions of a large set of  $X$  corresponding attributes--- hopefully including “typical” information---but have no prior knowledge of the kinds of, or number of, sample classes of which some of the attributes may be “typical”.

We will examine Case A here. Case B will be examined further on.

If no sample class is any “more similar” or “closer” to one class than to another (e.g., the classes are centered on points at the corners of an  $N$ -dimensional simplex;  $N = 3$ , an equilateral triangle;  $N = 4$ , a tetrahedron; etc.) then if we wish to assign a randomly drawn sample to its appropriate class, it will ordinarily require  $N$  operations of “comparison” to  $N$  different “type specimens” to match a sample to its class (i.e., to determine to which it is closest).

---

<sup>12</sup> (large compared to  $N$ )

<sup>13</sup> The effective “points of congruence” of a pair of raw patterns will operationally define which points (attributes) in one pattern (or list) “correspond” to those in another (49). This operation usually requires some prior simple normalizing preprocessing step. If more complex kinds of normalization or other types of preprocessing have been performed [for example, by arbitrary algorithms or by perceptrons (43)] which extract such features or attributes as the magnitude of “connectedness”, “convexity”, “circularity” etc. from each sample pattern and represent these as numbers, then the numbers representing the magnitudes of “connectedness” must be placed in corresponding positions on the lists or vectors representing those patterns; likewise for each other kind of feature or attribute.

If some of the classes are (or are defined as being) <sup>14</sup> "more similar" or "closer" to one class (real or artificial) <sup>14</sup> than to another, then the population of **N** classes can be subdivided (49, 18, 31) into **M** sub-classes such that members of a sub-class are, on the basis of "comparison", "more similar" or "closer" to one another (or to an arbitrary or nonarbitrary set of "type specimens") than to members of other sub-classes. (We can "discover" or "synthesize" "type specimens" for each of these sub-classes if need be. A "type specimen" may be equivalent to a page of our taxonomic key.) This process of subdivision may be further iterated on the sub-classes, and by this procedure we produce a hierarchical taxonomic tree.

If the number of subdivisions (branches) of the tree at each taxonomic level is **M** and if there are **L** taxonomic levels of branching above the Trunk of the tree., then  $N = M^L$ . It will take only **ML** sequential hierarchical operations of comparison, of a randomly selected sample to appropriate "type specimens" of the intermediate sub-classes, to match it to its proper class (branch tip). This is to be compared to the **N** operations for the unstructured case (e.g., the corners of the n-dimensional simplex).

Since  $N = M^L$ ,  $L = (\log_e N)/(\log_e M)$  and  $ML = M(\log_e N)/(\log_e M)$  .

The advantage in speed of a sequential hierarchic search over a straight sequential matching procedure is:

$$N/ML = N(\log_e M)/M(\log_e N) = N/M(\log_e M) ,$$

which is truly enormous when **N** is large and **M** is relatively small!

We wish to discover the value of **M** which minimizes **ML** and, therefore,

---

<sup>14</sup> It can easily be shown that even for the population with members that fall at the corners of the **N**-dimensional simplex there are **N** hyperplanes (each of which contains the centroid of the total population and bisects, at right angles, at least one line joining two vertices of the simplex) any one of which will equally sub-divide the population. The members of each sub-population (on each side of one such hyperplane) will be closer to the centroid of that sub-population (in such a case an unoccupied point, i.e., an "artificial" class) than to any members of the sub-population on the other side of the hyperplane. The resulting sub-populations can be further "bisected" iteratively. [If the two main branches are labelled (in binary notation) **1** and **0**, the sub-classes of **1** as **10** and **11**, and of **0**, as **00** and **01** respectively, etc., the set of binary numbers of the branch tips will constitute an efficient hierarchical nonsignificant Shannon-Fano Code (14,15).] There are at least  $N(N - 1)/2$  such distinct and equally efficient nonsignificant binary codes for each such population. Clearly, such a population can also be artificially divided into more than two parts at each step. Few if any natural populations will be as equally distributed in n-dimensional spaces as the vertices of simplexes. There will therefore be many fewer useful ways (but rarely only one way) to define closeness and to generate clusters. That is, there usually will exist more than one kind of efficient significant code that might be generated for natural populations.

maximize this advantage.

Setting  $(ML)/M = 0$  and solving for  $M$ , we find that  $e$  sub-classes per class would optimize  $M$ . Since the number of subdivisions must, in fact, be an integer, 3 subdivisions is optimal. (This is also the optimum for a relay tree memory.) However,  $(\log_e M)/M$  changes only slowly with  $M$  (see FIGURE 2) and, for example, the decimal taxonomic tree (or a decimal relay tree) will still be about 63% as efficient as a ternary tree.

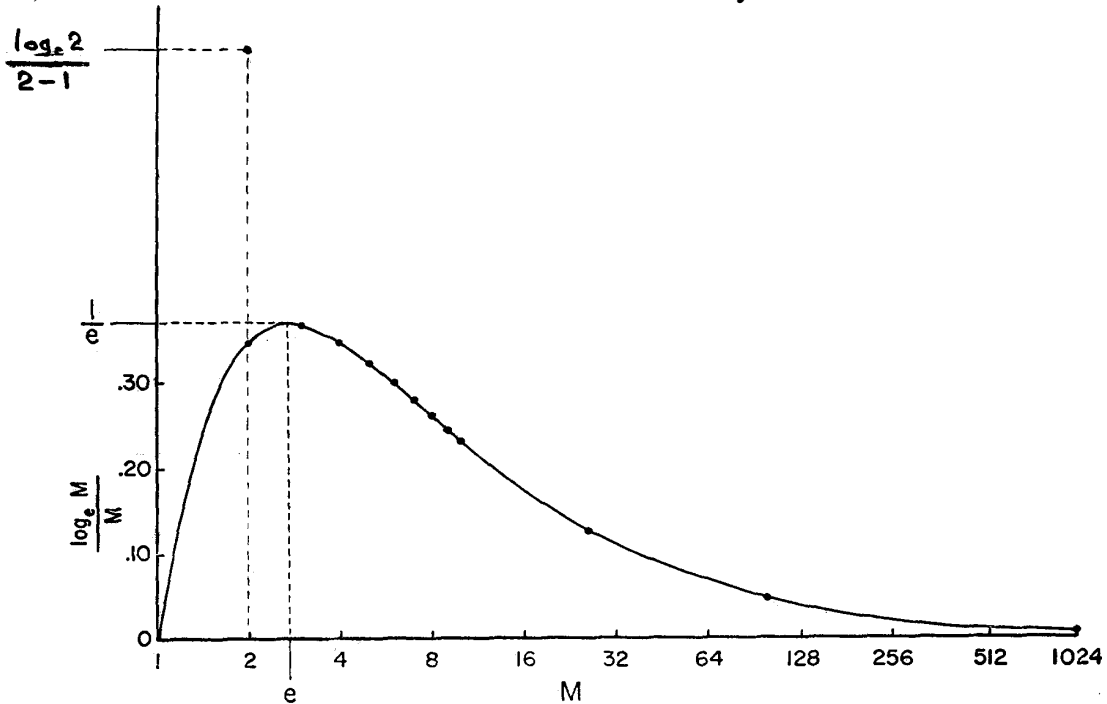


FIGURE 2

**FIGURE 2.**  $(\log_e M)/M$  vs.  $M$ , where  $M$  is the fixed number of branches at each branch point in a hierarchical tree. Efficiency remains reasonably high even out to  $M = 26$  (i.e., for alphabetic dictionaries).

If  $M = 2$ , and some threshold “similarity” or “distance” is arbitrarily measured with respect to only one of the two sub-classes (all samples closer than the threshold “distance” to that sub-class being assigned to it,--- all others being assigned to the “residue class”) then only  $M - 1$  operations of comparison are required at each level (i.e., we have one less degree of freedom in making class assignments) and the binary hierarchy then provides the most efficient sequential search because, although

$$N/(2\log_e N)/(\log_e 2) < N/(3\log_e N)/(\log_e 3) ,$$

after removing one degree of freedom,

$$N/(2 - 1)(\log_e N)/(\log_e 2) > N/(3\log_e N)/(\log_e 3) .$$

If the current in only one of each pair of branches in a binary relay tree is used to control the current in the other (as well as serving as part of a path to a branch tip), then only one external control line, controlling that branch at each level of the tree (instead of 2 control lines) [see (1)] will be needed to address a single branch tip, and the binary (rather than the ternary) relay or address tree then also becomes the optimum, in analogy to the case for the binary threshold classification, because again, one degree of freedom of choice has been eliminated.

It is worth noting, in passing, that this “proof” explicates conditions under which sequential binary logic [as opposed to ternary logic, see (24)] is most efficient. Also note, from **FIGURE 2**, that even alphabetic dictionary look-up (**M = 26**) is still 34% as efficient as the use of a ternary dictionary. Likewise, decimal page-number indexing (as noted previously) as well as decimal codes, such as the Dewey Decimal system for library indexing and mixed alphabetic and numeric codes, such as that of the Library of Congress, are also reasonably efficient significant hierarchic coding schemes for sequential retrieval---though again less efficient than the ternary, and still less efficient than binary threshold classification. The remarkable efficiency of the famous guessing game, Twenty Questions, also stems from the fact that an efficient taxonomic hierarchy will permit the identification or decoding of one in  $2^{20}$  ( $10^6$ ) objects or concepts with 20 well chosen yes-no questions.

The crux of the problem---what has previously been the “art” of classification---concerns the way in which one chooses well.

There are millions of species of organisms and thousands of different attributes that are used to develop taxonomic keys, yet only a very limited subset of these attributes need be examined in the classification of a single organism. And this provides the remarkable efficiency of well-constructed taxonomic keys. The magnitude of attention given to specific attributes can be thought of as weights, most of which are zero or near zero at most steps during the classification process, while a select few weights are uniquely large during particular steps. Biologists have given considerable explicit attention to the discovery and recognition of “weighty” attributes.

I believe the mental processes involved in arriving at a medical diagnosis are similar. However, less attention has so far been devoted to the

development of explicit medical diagnostic keys [see (3)]. Much is left to the development of “internal keys” during the continuing education of the physician. In the game of Twenty Questions, each participant uses his own internal (and often subconscious) “key”.

We will again return to these matters when we consider Case B, “Learning Without a Teacher”.

## VII Further Parallel Efficiency

Let us examine the added efficiency gained by processing the  $M$  pages at a single taxonomic level in parallel instead of sequentially. We will concern ourselves here with the total computational costs and real time to execute  $ML$  comparisons (measurements of “similarity”). If we use an additional central processing unit (CPU) for each of  $M$  parallel comparisons, the total machine-time costs will remain approximately constant, but execution in real time will be reduced almost by a factor  $M$  (at the same time capitalization must be increased by a factor  $M$ ). The larger than 2  $M$  is, the shorter the execution time. However as  $M$  increases, the time required to compare the computed “similarities” and, in turn, identify the largest among them (we call this the terminal comparison problem) grows from 0, when  $M = 2$  and one degree of freedom has been eliminated, to a time approaching that which it takes to measure a single “similarity”, when  $M$  approaches the number of attributes used in making such a “similarity” measurement.

We will now examine a class of binary hierarchic routines which, by resort to further parallel processing, can drastically reduce both of these latter execution times. Interestingly, the increase in efficiency again takes the approximate form of  $N/\log_2 N$ .

If we have a list of  $K$  terms on which we wish to perform a commutative algebraic operation (for example, addition), the machine time to complete the operation on a typical serial machine with a single CPU (arithmetic units and associated registers) will be proportional to  $K - 1$ . Suppose, instead, that we have a parallel machine with more than  $K/2$  CPU's, each with the speed of the CPU of the serial machine. We form  $K/2$  ordered pairs of the  $K$  serial terms, where  $K$  is even (or  $[(K - 1)/2] + 1$  “pairs” where  $K$  is odd) and perform the addition on each pair, form pairs of the resulting sums and iterate this procedure until the operation terminates with a single sum. This process will be  $(K - 1)/\log_2 K$  times faster than the addition on the simple serial machine, if  $K$  is a power of 2 [and  $(K - 1)/\log_2 L$  times faster if  $K$  is

not a power of 2 and  $\mathbf{L}$  is the next power of 2 larger than  $\mathbf{K}$ ].<sup>15</sup>

This procedure applies directly, with the same advantage, to the search for the largest member among a series of  $\mathbf{M}$  numbers (such as the terminal comparison problem) as well as to the determination of many measures of similarities between multidimensional vectors [e.g., Euclidian distances, Correlation Distance, Tanimoto Distance (49)] such as the measure of similarity of the set of attributes of the unknown to those sets of attributes which are listed on each of the 10 pages at a branch point in the decimal taxonomic key.

The useful limit to which combinations of these various devices of parallel processing can be pushed, by increasing  $\mathbf{M}$  as well as increasing the parallelism within each of the  $\mathbf{M}$  parallel processing units, is finally set by the maximum achievable rate of transfer of blocks of pages between memory levels, or alternatively, by the cost of an adequately large and fast true random-access mass-memory. As the number of parallel CPU's increases, the complexity of the time-sharing software and hardware packages that would have to monitor these multi-processing routines increases even faster, and this realization has previously inhibited serious attempts at the development of large-scale parallelism in general purpose machines. However, the very recent and growing numbers of successes in the development and installation of time-sharing systems together with increasing appreciation of the potentials of parallel machines (56b), may help to stimulate new efforts along these lines.

## VIII The Classification of Monosyllabic Speech as a Model

The relationship of the previous discussions of hierarchic efficiencies to the efficient real-time coding and decoding of speech will become apparent in the following section, where we will also examine operational problems of “learning without a teacher”, of generating an efficient code, of defining “typicality” and of learning with a teacher. The data set chosen for exploring our model is useful and is sufficiently complex <sup>16</sup> to require a reasonable amount of attention to details. Relevant aspects of speech have been fairly well studied (17) and may be familiar enough to the reader so that experience and intuition can be expected to aid materially in evaluation

---

<sup>15</sup> A similar principle is also responsible for the efficiency [  $N/(2\log_2 N)$  ] of the now famous Fast Fourier Transform technique of Cooley and Tukey (10), and its derivatives (e.g., 56a).

<sup>16</sup> and yet is not as demanding of as elaborate preprocessing and normalizing for reduction of dimensionality (49) as would be required, for example, for detecting invariances for the pattern-recognition of high resolution television images of our environment.

of this model.

For some time, considerable attention has been given to the possibilities of recognizing speech, phoneme by phoneme. If the 39 or so phonemes of American-English were essentially invariant, independent of the preceding and succeeding phonemes (i.e., independent of context), a taxonomic “phonetic dictionary” with about 75 pages would permit coding of speech over a channel with a capacity of about 60 bits per second (41). Unfortunately, phonetic context leaves phonemes less invariant than syllables (34,47a) and therefore the syllable approach, even though it requires a much larger dictionary or key, warrants careful attention.

Connected natural speech averages about 2.6 syllables per second. 1,370 syllables account for 93.4% of typical written American-English (12) and a total of about 4,500 syllables can account for essentially all of American-English (41). If all were equally frequent, the 4,500 syllables could, in principle, be coded by about 12.1 bits each. If the known frequencies of occurrence of syllables is plotted as a function of the reciprocal of the rank order of frequency (i.e., frequency vs. 1/rank order) in common with such plots of the words in a single user’s vocabulary in a single language, they fall on a straight line with a slope which is a function of the vocabulary size (71, 55). A vocabulary with such a distribution provides only about 9.2 bits of information per syllable (41) and therefore, only an average of about 24 bits per second should be required to code connected monotone speech of a single speaker, syllable by syllable.

In general, if we choose to code directly for larger and larger segments of messages (for example, words, phrases, sentences, etc.) the size of our dictionary grows rapidly, and the potential coding efficiency increases.<sup>17</sup> For example, if we chose to code for all phonetically distinguishable whole words in spoken American-English, the number of dictionary items would probably be fewer than  $2^{20}$ , and the potential coding efficiency would permit utilization of channels with capacities smaller than 24 bits per sec.. The limit of coding efficiency is generally approached asymptotically, as one codes for larger and larger segments of a message (59). This means, of course, that larger and larger dictionaries are favored less and less. Language evolution may reflect this fact. Whereas new words are added to an established language relatively infrequently, “new” pairs of words must occur quite frequently. This is because the grammatical and syntactical

---

<sup>17</sup> ...because of the changes in frequencies reflected by the change in slope of the frequency vs.1/rank order plot. As a result, even if phonemes were invariant, independent of context, coding by syllables would be 2.5 times more efficient (i.e., 60 bits/sec. divided by 24 bits/sec.).

rules of natural languages permit the generation and decoding of messages, some parts of which (as well as the whole) are likely to be unique in the experience of the speaker (or writer) as well as in the experience of the listener (or reader). Were this not the case, then impossibly <sup>18</sup> large dictionaries would be required to decode messages. Since the transition to almost total dependence on grammar and syntax occurs at about the word level of segmentation, the bulk of the potential for channel capacity reduction by use of a large dictionary can probably be harvested at or near the syllable level of segmentation of spoken messages.

I will therefore assume that the bulk of the central problem of efficient coding and decoding of American-English speech can be reduced to the problem of the automatic recognition and coding of monotone monosyllables of a single speaker. The problems of automatic segmentation, of automatically dealing with intonation, stress, speed of speaking, loudness, and other prosodic cues, as well as dialects and individual idiosyncracies, are by no means trivial, nor have they so far been adequately solved; however, by setting these problems aside, it will be easier to see the shape of the solution to the central coding problem.

How might we code monosyllables at about 9.2 bits per syllable?

The detailed wave forms, at the output of a microphone, of two consecutive utterances of equal duration and loudness of the same monosyllable by the same speaker, while displaying very many features in common, in general will show a poor correspondence when points in the two waveforms are best-matched on a time coordinated basis. This results from very slight and often nonlinear phase shifts between the two records. Since a listener is usually unable to detect such differences monaurally, they are not considered to be information-bearing differences of significance in telephone applications.

The classic work at the Haskins Laboratories with the “pattern play-back

---

<sup>18</sup> Suppose that there were  $2^{20}$  phonetically distinguishable words in American-English and we chose to generate a “dictionary” of all permutations of all groups of 10 words. If all these combinations were equally likely, a dictionary with a few less than  $2^{200}$  entries would be required. If the frequencies of occurrence of the combinations followed the same law as syllables and words (see page 24), then only an average of 100 bits would be required to code for a sequence of 10 words (55) providing a saving of less than a factor of 2 over the syllable code. Note, however, that since there are supposedly about  $10^{79} = 2^{182}$  protons and neutrons in the observable universe, insufficient matter exists to permit the construction of a dictionary of this size.

machine” (11) has demonstrated that the so-called short-time power spectrum of speech which does not preserve the phase information, nonetheless preserves the information content for monotone reproduction of speech. The “spectrum of the log-spectrum” [which Tukey has named the cepstrum (7)] permits one to “recover” the fundamental frequency of the voiced parts of speech, even when the fundamental falls below the lower boundary of bandlimited speech, as is usually the case in telephony (45,46). In combination, these two kinds of transformations provide sufficient information to permit regeneration of a quality replica of the original non-monotone speech (48).

We will assume that each monosyllable of speech has been transformed by passing it through a telephone and then through 2 real-time hybrid spectrum analyzers such as the Federal Scientific UA-7 (65), measuring the power in 1.5 db. steps on one analyzer with 250 hz. bandwidth in the interval 250 to 3,000 hz., and 8 millisecs. resolution along the time axis, and on the other analyzer with 50 hz. and 40 millisecs. bandwidths. We will also assume that the spectra from the first of these are sampled in 20 contiguous 125 hz. bands every 16 millisecs. and the spectra from the second are passed through a third such analyzer with 20 contiguous 125 hz. bands, sampled every 16 millisecs. to produce a cepstrum. The analogue outputs of the first and third analyzer are digitized and are recorded as 2 time-coordinated lists (vectors) of, on the average, 400 numbers, (6 bits per number)---producing an average of about 4,800 data bits per syllable. In contrast to the case for untransformed waveforms, transformed patterns of consecutive utterances of the same syllable by the same speaker match very well. (If the spectra of a syllable voiced by different speakers are normalized by amplitude, time (58) and frequency (20) linear compression or expansion, the match is still quite good, and generally appears much better than the best match between two different syllables.)

With our model, at least 4,500 pages, 4,800 bits/page (a total of  $2.2 \times 10^7$  bits) would be needed for the phonetic taxonomic key to code and decode individual monotone syllables of American-English speech efficiently. Although procedures will be considered for somewhat reducing the memory needs, the scale of this kind of problem begins to justify, I hope, the attention given, in the previous discussions, to the harnessing of hierarchic efficiency, and hierarchic structuring of memory, for economic real-time searches.

We will next examine some processes for generating such a key, first intuitively and then somewhat more operationally:

Suppose we were to listen to the 4,500 monotonic syllables, presented at random, and with frequencies of presentation of the individual syllables equal to their frequencies of occurrence in connected American-English, and suppose, further, that we are able to divide the set into 2 almost equal sub-sets such that, on the average, the members within a sub-set “sounded” more similar to one another than to members of the other sub-set, numbering the larger sub-set **0** and the other **1**. We now subdivide each sub-set in half, again on the basis of “phonetic similarity”, and continue the procedure until we have the 4,500 branch tips, each with something like a 12-bit binary label. This hierarchical collection of 4,500 12-bit binary labels and the associated 11-bit, 10-bit,.....1-bit labels, for the intermediate branches, would constitute part of a fairly efficient phonetic dictionary and code book for a Significant Shannon- Fano code (49).

We will now outline a set of “educational” algorithms which might be expected automatically to generate something like a phonetic dictionary and hierarchical code with many of the desired properties [e.g., it should use nearer to an average of 9.2 rather than 12 binary digits per coded syllable; generates a comma-free code (14); can be used for decoding as readily as for coding; and code-word "prefixes" are meaningfully decoded with the same dictionary,---i.e., it is a significant taxonomic code.

The learning procedure, like most "educational experiences", will consume considerable amounts of (machine) time and actually need not operate "on-line". However, once the key or dictionary has been generated, it would be used on-line and in real-time. Thereafter, only aspects of updating [see (49)] would normally be performed off-line.

The problem of efficient coding of monosyllabic speech has previously received some attention in the literature (e.g., 47a,47b,56c). Those approaches parallel that outlined below in a few details, but have not been hierarchic nor have they been described in a way that might have general applicability in other pattern-recognition problems.

## **IX Learning Without a Teacher, (Case B)**

1). 4,500 syllables, with their natural frequencies of occurrence, are sampled at random, and for each syllable the 2 time-coordinated lists of up to 400 6-bit numbers are read. The cepstrum list is first examined, 16 millisecond time-frame by time-frame along the “frequency-equivalent” axis <sup>19</sup>, in order to detect the times of appearance and disappearance of signals

---

<sup>19</sup> Tukey has called this dimension of the cepstrum the quefrequency (7).

associated with the fundamental resonances of the vocal chords (i.e., the beginning and ending of so-called voiced portions of the syllable, including a vowel and any contiguous voiced consonants such as j, l, m, n, r, v, and z). The appearance of energy in the band corresponding to 90 - 300 hz. indicates voicing. The time-frame that is closest to the center of the voiced interval is arbitrarily designated as the "reference time-frame",  $t = 0$ . Each of the 2 lists of numbers for each syllable is loaded into memory with its reference time-frame placed at the "central address" in its own 2,400-bit half of a 4,800-bit block of memory. This means, for example, that the numbers representing the sounds for "end" in the words, **end**, **bend**, and **send**, will be "aligned". [However, they will be usually somewhat out of phonetic alignment with **sends**, **bends**, as well as with **rend** and **lend** because the closing s sound in the first case, and the preceding r and l sounds in the second case, are parts of the voiced intervals (see **FIGURE 3**).]

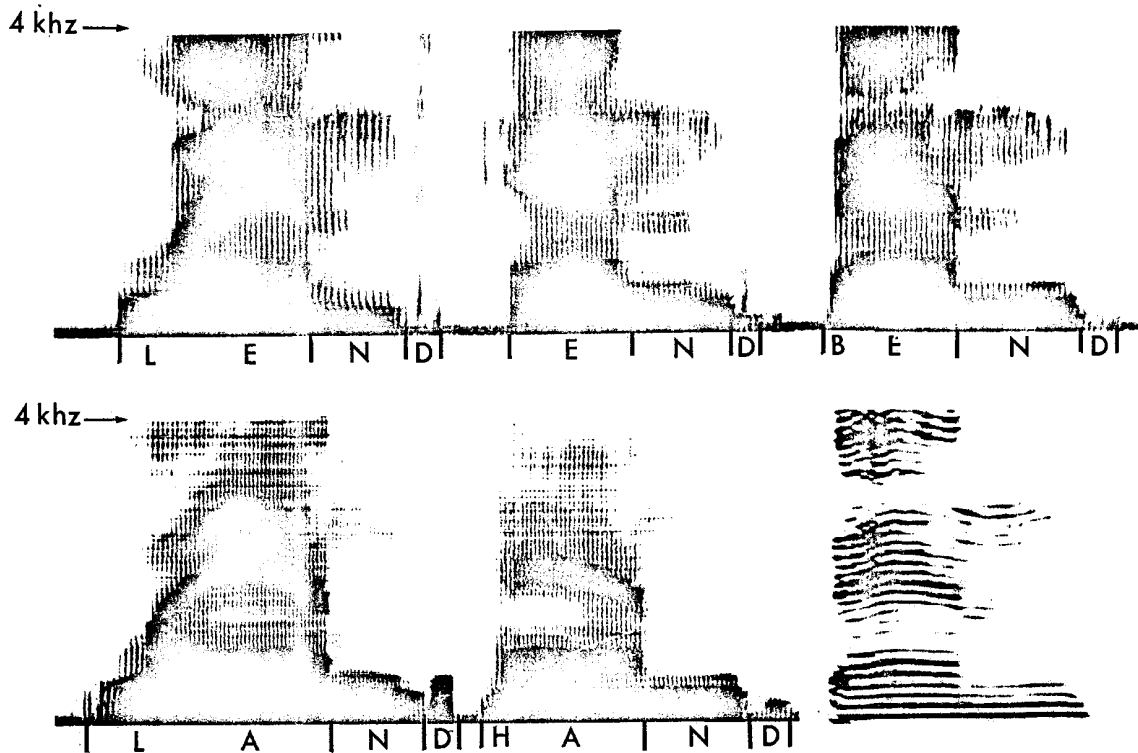


FIGURE 3

**FIGURE 3.** Frequency vs. time. The short-time spectra of 5 isolated syllables were recorded with a frequency resolution of 300 hz. and a time resolution of 7.2 milliseccs, and one syllable (**BEND**) was also recorded with spectral and time bandwidths of 45 hz. and 48 milliseccs (lower right). Note the approximately constant duration of the voiced "N" interval in all 5 syllables, the longer duration of the "A" as compared to the "E", and the appreciable overlap of the voiced "L"-vowel transitions. In the narrow-band spectrum, the harmonics of the fundamental frequency are readily apparent during voicing. It is these harmonics that provide the information which permits the recovery of the fundamental frequency in the cepstrum.

2). When a sample of statistically significant size has been collected [see (49) for examples of explicit algorithms for making this decision] a set of measures of similarity or “distance” [for example, one of the following: Tanimoto Distance,<sup>20</sup> Correlation Distance and less usefully, Euclidean Distance, etc. (49,31)] is computed between an 800-dimensional 4,800-bit vector, representing an arbitrarily chosen syllable, and each of the vectors representing the other syllables in the sample. The histogram of these measures (frequency vs. similarity) is examined, the “largest peak” [see (49)] is located and

3). a syllable “at that peak” is selected.

4). The set of similarities between this syllable <sup>21</sup> and the others is again computed, and the new histogram is examined. If the largest peak has now moved “sufficiently close” [see (49)] to the origin <sup>22</sup> of the histogram, we go on to 5). Otherwise, we repeat 3).

5). We now compute the similarity between this “peak syllable” and a second syllable, shifting the registration of the reference time-frame of the second syllable, to search for that alignment which gives the greatest similarity.<sup>23</sup> The second syllable is returned to its memory block but with a new reference time-frame, namely that which gives it a “best-match” with the peak syllable. This process is repeated for all the remaining syllables. A histogram of the best-match similarities is examined and, if the largest peak is sufficiently close to the origin, we go to 6). Otherwise, we repeat 3). (Note, at this stage, if the peak syllable were, for example, “end”, misalignment, such as with **rend**, **bends**, and **lend** would have been eliminated.)

---

<sup>20</sup> The Tanimoto Similarity Coefficient for continuous variables is defined as follows: For 2 vectors,  $x_1, x_2, x_3, \dots, x_i$  and  $y_1, y_2, y_3, \dots, y_i$ , compare each  $x_i$  with each  $y_i$ . When  $x_i$  is equal to or greater than  $y_i$  or alternatively, when  $y_i$  is greater than  $x_i$  assign  $x_i$  or in the alternate case,  $y_i$  to class **A**. When  $x_i$  is less than  $y_i$  or alternatively, when  $y_i$  is equal to or less than  $x_i$ , assign  $x_i$  or in the alternate case,  $y_i$  to class **B**. The sum of the members of class **B** divided by the sum of the members of class **A** equals the Tanimoto Similarity Coefficient, **T**. **Tanimoto Distance** =  $-\log_2 T$ , which is equal to 0 for identical vectors and is infinite for completely disjoint vectors (49).

<sup>21</sup> (which now is likely to be more “representative” of a large fraction of the sample than the first arbitrarily chosen syllable).

<sup>22</sup> For a similarity measure, the origin is at similarity = 1; for a “distance” measure, the origin is at distance = 0.

<sup>23</sup> Note, a hierarchical search will substantially improve speed in this operation as well.

6). A table is now prepared (in memory) where the 800 rows correspond to the 800 numbers representing the attributes of each syllable (aligned as the consequence of 5) and the 64 columns of the table contain  $\mathbf{f}_{r,t}$ , the frequencies of occurrence of the various 6-bit numbers representing the power (or log power) in each of the corresponding frequency (hz.) or quefrequency bands in each time-frame of the syllable vectors (i.e., each row is identified uniquely by two subscripts,  $r$ , designating frequency or quefrequency, and  $t$ , time-frame). Next, each number (which is stored as one of the 800 6-bit words that represent a syllable) is “weighted” by multiplying it by the weighted self-information for that entry,  $-\mathbf{f}_{r,t} \log_2 \mathbf{f}_{r,t}$ , and the new set of numbers is stored in a second portion of memory in the same format as the unweighted data set. (Note, if identical values were to occur in the corresponding time-frames in all syllables----for example, a value of 53 in the 250 - 375 hz. band in all the reference time-frames----then, since  $\mathbf{f}_{250,0} = 1$ , those weighted numbers will be 0.) As  $\mathbf{f}$  approaches 0, the weighted number will also approach 0. On the other hand, when  $\mathbf{f}_{r,t} = 1/e$ , the weighting will be maximal. In fact, the weighting varies as does  $(1/M)\log_e M$  (FIGURE 2) and therefore attribute magnitudes which are “favored” by this weighting procedure occur frequently enough, but not too frequently, so that subdivision of the sample into an efficient number of branches is subsequently favored.

We now compute the similarities between the weighted peak syllable and the other weighted syllables, and if the largest peak is sufficiently close to the origin, we name the peak syllable the archetype for the “First Main Branch” (Branch 0) and go to 7). If not, we select a syllable at the largest peak, clear the frequency table and that portion of the memory which contains the weighted data and go back to 4).

7). We next locate that minimum, in the last histogram, which most nearly and cleanly divides the population in half [see (49)] .

8). We transfer the archetype of Branch 0, and the samples closer to it than  $\mathbf{d}_{\min_0}$  (the distance to the minimum) from the original memory locations to new locations and clear the original locations.

9). The original data on the archetype itself, the frequency table for the Tree Trunk and  $\mathbf{d}_{\min_0}$  are all transferred to page 0 (in binary notation) of the “taxonomic phonetic key”. The remaining syllables in the original locations are members of the “First Residue Branch” (Branch 1). [When step 8) has been successfully completed on Branch 1, the original data on the archetype for the “First Main Branch of Branch 1” (Branch 10), the frequency table for Branch 1, and  $\mathbf{d}_{\min_{10}}$  are transferred to page 10 (in

binary notation) of the taxonomic phonetic key.]

10a). We next check for the statistical significance of the size of the sample remaining in the Residue Branch. If the size is significant, we select a sample which is located at the largest remaining peak of the last histogram and go to 4). If not, go to 11).

10b). We check for the statistical significance of the size of the sample in this Main Branch. If the size is significant, we go to 2). If not, go to 11).

11). We take a new syllable, locate the reference time-frame as in 1), and turn to page **0** of the key and frame-shift to get the best match with the original data set for the archetype; weight the new syllable and the archetype using the frequency table on page **0**; measure the similarity between the two, and assign the syllable to the original data set of this branch (in its new location in memory) if the distance is less than  $d_{\min 0}$  and go to 10b). If not, we add the syllable to the Residue Branch at this taxonomic level and go to 10a). In this way, samples will accumulate in sets **0**, **10** and **11** (**11** is the as yet unprocessed Residue Branch of the First Residue Branch). Whenever the test of significance of sample size is satisfied for one of the sub-sets, the steps from 10) on are repeated. [Note, because only one comparison needs to be made at each level of the binary tree, Residue pages, **1**, **01**, **11**.... (i.e., all pages with numbers ending in a **1**) can be left out of the key.]

In this way, we would automatically generate a phonetic taxonomic key as well as an automatic procedure for coding any input syllable. Because the frequency of occurrence of syllables affects which peaks are the largest, the code generated will be more efficient than that which would be generated for 4,500 equally frequent syllables (i.e., will use less than an average of 12.1 bits/syllable). After transmission, a decoding procedure based on the use of the same dictionary and using the stored attributes of the syllables as control signals for “resynthesis by rule” (38,22,54) would complete the telephone transmission sequence.

Note that for telephone use, we do not require “perfect” identification of individual syllables. The identification of isolated monosyllables is much more difficult for a human than identification in context in connected speech (17), (for example, distinguishing “**lends**” from “**lens**”). Since the ultimate user of the coded and decoded syllables will receive them in a reconstruction of connected speech, the precision of identification required in that context is less demanding than for a voice-operated typewriter or voice-input to a computer, where some grammatical and syntactic editing

by the machine (and/or the speaker) would usually be essential.

The key generated by the above rules requires many more bits per page than was indicated earlier because of the space requirements for the large frequency tables. In the following discussion of typicality, devices for substantially reducing this memory requirement will become apparent.

## X Typicality

Let us suppose that the above rules were applied to the data set representing the following syllables:

**and, end, band, bend, bands, bends, bland, blend, blends, canned, dand, dend, fend, fanned, fends, gand, hand, hands, land, lend, lands, lends, manned, mend, mends, panned, penned, pends, rend, rends, sand, send, sands, sends, spanned, spend, spends, tanned, tend, tends.**

The syllables with the “**end**” root are more numerous than those with the “**and**” root. Therefore, it is to be expected that the “largest peak” would contain the “**end**” group. The alignment procedure would result in fairly good lining up of the time-frames for the “**nd**” sound in all the syllables, and since this sound is common to all the syllables, the weighting step would reduce most of the numbers representing the utterance in the “**nd**” time-frames to very near zero. The numbers representing the disjoint parts of the two vowels (which are each present in about half of this sample of syllables) will receive the highest weight and, since the power level during voicing is high, the effect of the vowels will dominate the computation of the similarity measure. Subdivision, [step 8)], will produce two sets, the “**end**” family and the “**and**” family. Further processing of the “**end**” family will result in weighting which would reduce the numbers in the “**end**” time-frames to near 0. In terms of these processing procedures, attributes which receive high weight at a given level of the taxonomy and low weight at the very next level are typical of that next level. Thus, if “**nd**” had received high weight at the level next closest to the “syllable trunk”, the “**nd**” sound would be typical of the “**and-end**” group, whereas the vowel sounds would each be typical, respectively, of each of the two subsets generated by step 8).

If we sum the weighting terms for a row of attributes, we compute the average self-information conveyed by that attribute in that population (**A**),

$$\mathbf{H}_{r,t_A} = - \mathbf{f}_{r,t_A} \log_2 \mathbf{f}_{r,t_A} .$$

If we perform the same operation for the same <sup>24</sup> attribute in a branch at the next level, (**B**), away from the trunk of the tree, and the change in the average self-information,  $\mathbf{H}_{r,t_A} - \mathbf{H}_{r,t_B}$ , is near one bit, then this identifies a typical attribute.

This points the way for substantially reducing the space in memory needed for each page of our phonetic taxonomic key.

Note that it is the automatic increasing sensitivity to typicality that results from the fresh renormalization (in this case, translational renormalization along the time axis, using the reference time-frame of the archetype,----step 5), followed by the informational reweighting, step 6), after each hierarchical subdivision, that gives this kind of classification (49) far more power than any other kinds of hierarchical or non-hierarchical classification (4,27,40). The essentials of this process are not entirely novel. Double-beam photometry on the one hand, and the use of differential amplifiers, on the other, to ratio-out or subtract out redundant “common-mode noise”, constitute rather familiar forms of “informational reweighting or filtering” (49) of signals. This kind of process is generalized here and is iterated at each hierarchical level. It is useful to visualize this renormalization and reweighting as a means for selectively “distorting” the nested sets of n-dimensional spaces occupied by the samples so as to selectively increase the separation between the sub-groups [see (49) and (31)].

To divide the “**and-end**” population of syllables, actually only the address of one attribute in a vowel time-frame (the address, measured from the archetype’s reference time-frame) and the archetype’s entry at the same address, would be required. These could substitute for both the entire frequency table and the listing of the full set of attributes of the archetype, because, for the purposes of this subdivision, all remaining bits are redundant or noise. In general,  $\mathbf{H}_{r,t_A} - \mathbf{H}_{r,t_B}$  for typical attributes will carry something less than 1 bit of information (49); therefore the addresses of a number of only partially correlated attributes, the applicable portions of the frequency table, and the archetype’s entries at the same addresses will have to be carried on a page of the key. In addition, a new  $\mathbf{d}_{\min}$ , representing the location of the minimum in the histogram, recomputed using the truncated data set, would also appear on the page.

---

<sup>24</sup> Note that because of possible shifts in alignment of reference time-frames generated by step 5), as one moves from level to level, the term, “same”, may appear somewhat ambiguous. For our purposes, the equivalence of time-frames at two adjacent levels of a tree is determined by aligning the reference time-frames of the two pertinent archetypes (and this will always be the original reference time-frames).

Just which addresses to use to designate the typical attributes for the key can be determined as follows:

We compute the similarities among that set of attributes (not syllables) which show differences in  $\mathbf{H}_{r,t_A} - \mathbf{H}_{r,t_B}$  greater than some arbitrary amount (for example, 0.25 bits), however, we replace each original datum by its self-information,  $-\log_2 \mathbf{f}_{r,t}$ , before computing the similarities. [For the “**and-end**” set of syllables, many entries for many time-frames in the “**nd**” region would be found to be “identical”---i.e., redundant (neglecting the effects of noise).] As entries for the key, we select an address-interval and the magnitudes of the attributes at the time-and frequency (or quefrequency) centers of each of those regions (or sets of attributes) which show some arbitrarily high level of similarity to one another within a set, but show lower similarity between sets. This provides some redundancy and some protection against reference time-frame (and frequency or quefrequency) misalignment.

Other sophisticated automatic methods for reduction of dimensionality (61) may also be used to determine which are the information-bearing typical attributes.

All of this time-consuming processing for preparing the reference file or dictionary is part of the “education period” of a machine. This kind of off-line processing, however, should permit a machine to learn to ultimately perform syllable recognition on-line and in real-time, if large scale parallelism and large fast hierarchical memories are developed. Such hardware would allow efficient parallel search of keys in which  $\mathbf{M} \gg 2$  (see page 21). By adding sophisticated segmenting (56) and normalizing techniques, and by increasing the dictionary size to about  $10^6$  pages to accommodate various stressed or accented forms of the syllables, I believe ordinary connected speech would be automatically codable at about 40 bits/sec. in real time. If the dictionary were made still larger, quality sufficient for speaker recognition should be possible, still with substantial conservation of channel capacity. A user, by designating the quality desired for a call (probably when dialing) would buy just the channel capacity he needs. The dialing code for a minimal-cost call might, for example, provide instructions for coding and decoding each syllable to 9 bits; that for a high-fidelity call, 40 bits, but only one dictionary would be required for these two kinds (or for any intermediate kinds) of service. This is one of the dividends of a Significant Taxonomic Shannon-Fano Code [see page 8 and (49)].

## XI Learning With a Teacher

For some learning tasks, intervention of a teacher may not only be unnecessary, but may even be undesirable in that prejudices of the teacher may be confining. For example, the experienced clinician who “knows all diseases” (a fiction) in teaching his students, may “help” them to overlook separate disease entities which are classically lumped under one name.

However, in the syllable recognition task, we know that we want a code which is significant, not only from the machine coding point of view, but also from a human user’s frame of reference, because the machine is intended to intervene between a human speaker and human (or machine?) listener. When the communication channel is noisy and the machine therefore “makes a mistake”, we would like it to be the kind of mistake that a human might make (for example, hearing “**brand**” as “**band**”) so that the human listener will have the opportunity for making the usual contextual corrections. We would, therefore, examine the tree (for example, moving from branch-tips down) and, where the divisions seem “phonetically unnatural”, modify the arbitrary elements in the program (i.e., the choice of similarity measure, the measure of significance of sample size, the definitions of largest peak, of minimum, etc.) to try to force a division which “seems” more natural. Such “teaching procedures” merge with program-debugging, [i.e., the examination of (arbitrary) coding steps to see if they, in fact, lead to the intended goal]. The arbitrary aspect of the judgement of what “seems” more natural, can be made more objective by using confusion matrices, generated by polling large numbers of listeners, and then applying Shepard’s techniques for reducing dimensionality within such matrices (60,61,53) to generate “more natural classes” as frames of reference for “teaching”.

## XII Medical Classification

Natural languages seem to evolve in ways that preserve reasonably large phonetic “gaps” between most words (gaps in “phonetic space”, not in connected speech). As a result, the vocabulary of American-English (well under  $2^{20}$  words) occupies a minute fraction of the total possible phonetic space of a telephone channel (at about 15,000 bits per word, there would be  $2^{15,000}$  possible distinguishable “words”). The evolution of organisms, through so-called species isolating mechanisms, also assures fairly large gaps between most species of organisms. Most conceivable hybrid organisms of the sort dog-elephant, do not exist. Likewise most possible “hybrid words” like dophant or gelph, are not found in English dictionaries .

The situation appears to be somewhat different in the multi-dimensional space occupied by human diseases. Patients can, and occasionally do, suffer from both diabetes and anemia; heart disease and lung cancer; duodenal ulcer and pneumonia , etc., and with varying degrees of severity of each of the members of the disease pairs. “Disease Space” therefore, does not necessarily contain the large empty gaps between clusters that make “natural” subdivision (49) easy. Since it would not be adequate if an automated diagnosis machine were only able to recognize one out of three or four diseases from which a patient may suffer, a change in approach is required. By modifying step 8) in the following kind (49) of way, the problems just discussed generated by “fuzzy sets” (70,6), are largely solved:

We transfer the archetype of Branch **X** and the samples closer to it than  $\mathbf{d}_{\min_x} + \mathbf{Y}$  from the original memory locations to new locations, and clear all original locations occupied by samples closer than  $\mathbf{d}_{\min_x} - \mathbf{Y}$  to the archetype. [Changes in 8) underscored]

This modified form of step 8), assures that at each subdivision, that portion of the population in the interval,  $\mathbf{d}_{\min_x} \pm \mathbf{Y}$ , will appear in both sub-groups. Let us suppose that 10% of the population falls in this overlap group at each subdivision, and that our trunk population consists of 4,500 equiprobable kinds of syllables. Then the taxonomic tree generated with modified step 8) would have about 4 x 4,500 pages and about 12 + 2 levels.<sup>25</sup> The code generated would therefore require an average of about 2 bits more per syllable than if unmodified step 8) had been used, and, on the average, each syllable would be represented by about four different code words.

In this way, the model is easily extended to deal usefully with fuzzy sets such as those to be encountered in medical diagnosis. In some ways fewer processing problems will be posed but as yet, usually more data acquisition problems will have to be faced. For example, the attributes representing blood sugar level and mean red cell volume could each be regularly assigned its own standard address in the vectors “describing” patients; therefore frame-shift techniques should not be required to align vectors. In the course of acquiring data from typical examinations of a patient, “questioning” usually follows a hierarchical procedure, so that the set of questions asked (and tests and therapies ordered) is far smaller than the total number of possible questions and possible tests and therapies. This approach is

---

<sup>25</sup> The number of levels produced by modified step 8) will be approximately equal to  $\log_2 N[1 + \log_2(1 + \mathcal{X})]$ , for  $\mathcal{X} < 0.2$ , where  $\mathcal{X}$  is the fraction of the population that falls in the overlap region.

efficient because the physician already has developed his internal dictionary and, in a sense, is using coding procedures like those outlined for machine syllable recognition and, at any one time, is concentrating only on those attributes which are likely to be typical at the next level of the tree. But if the machine is to develop its own “medical dictionary” and therapeutic index, without a teacher, a very large set of standardized questions (and tests) must be asked of a very large number of patients during the educational period. Uniform and acceptable protocols for the collection of the data of physical examinations, laboratory tests and histories, adequate for this educational process, have yet to be developed (5,39).

Increasing numbers of clinical chemical techniques (for previously unmeasurable physiological substances) which are easily adaptable to automated clinical laboratory instrumentation (66) are under development. The main interest in such efforts stems from the hope to turn up tests which might provide new information to supplement the present diagnostic armamenta of the physician. However, many of these may well prove to be redundant in the sense that they will be “typical” of disease states which are already indicated by observations made, for example, during the physical examination of the patient. This also means that they could “substitute for” such physical observations in the vectors describing patients just as the differences in 1) electrophoretic mobilities among proteins; 2) biochemical metabolic pathways; 3) amino acid sequences in proteins such as the hemoglobins or cytochromes; and 4) homologous base sequences in nucleic acids (e.g., as are regularly described in such journals as Comparative Biochemistry and Physiology) begin to permit us to classify organisms as efficiently as was previously only possible by physical examination of the whole organism. Therefore, even such redundant tests will be valuable in the context of automated diagnosis because, if this technology develops sufficiently rapidly, the 100- to 1,000-dimensional vector output for each patient, automatically generated by such clinical laboratory machines (to be compared to the 6 to 20 at present) may reduce the magnitude of the effort required to prepare the data from physical examinations and history taking for the education of the computer.

When such problems have largely been solved, my guess is that completely automated medical diagnosis will have approached a level of technical performance comparable to that achieved about 6 years ago by programs for the automatic processing of technical Russian texts. Inelegant but usable translations into English were produced (47), but even more importantly, that kind of performance provided additional stimulus for further productive linguistic research as well as further successful translation software development (69). Continuing parallel developments in

automatic semantic information processing (42a) at higher levels of the semantic hierarchy than that of “phonetic meaning” reviewed in some detail here, should aid substantially in the application of artificial intelligence to the automation of medical diagnosis. The changes in medicine that will almost certainly be stimulated by the early attempts to make diagnosis explicit enough to be managed by simple-minded machines are likely to be far more pervasive and profound than those produced as a direct result of early machine successes at automated diagnosis.

In the meantime, an off-line syllable recognition program may itself have medical potential for the study of certain aphasias and other “speech defects” related to brain “damage”.

## SUMMARY

Properties of taxonomic keys and hierarchical logic which permit the design of search techniques with increased efficiency, over simple sequential searches, of  $N/M \log_M N$  (where  $M$  is the number of branches at each branch point in a taxonomic tree, and  $N$  is the total number of branch tips---i.e., item classes) have been examined. Near optimum significant codes which can be produced with such techniques have been discussed. Hierarchic hardware configurations for mass-memories together with hierarchic parallel processing techniques, which can provide additional speed-up factors of  $M$  and  $K/\log_2 K$  (where  $K$  is the number of sequential algebraic operations performed on a list with a serial computer) have also been reviewed. These will permit such coding and decoding in real-time. For purposes of illustration, algorithms for a hierarchical pattern-recognition model for automatic speech recognition (without a teacher) have been outlined. Such techniques should permit transmission of speech over channels with capacities as small as 50 bits/ sec. Relevance to automated medical diagnosis has also been briefly discussed.

## **BIBLIOGRAPHY**

1. Ahrons, R.W., *Superconductor Circuits* , U.S. Patent **3,207,921** , 1965.
2. Anderson, L.K., Holographic optical memory for bulk data storage. *Bell Labs, Record* **46**; 319, 1968.
3. Atamer, M.A., *Blood Diseases*. Grune Stratton, New York, 1963.
4. Baron, D.N., and Fraser, P.M. Medical applications of taxonomic methods. *Brit. Med. Bull.* **24**:236, 1968.

5. Bates, J.A.V. Preparation of clinical data for computers. *Brit. Med. Bull.* **24**:199, 1968.
6. Bellman, R., Kalaba, R., and Zadeh, L.A., Abstraction and pattern classification. *J.Math. Anal.* **13**:1, 1966.
7. Bogert, B.P., Healy, M.J.R., and Tukey, J.W., The quefrency analysis of time series for echoes, cepstrum, pseudo-autocovariance, cross-cepstrum and saphe cracking. *Proc. Sym. on Time Series Analysis*, M. Rosenblatt, ed., Wiley, New York, 1963, pp. 209-243.
8. Brenneinan, A.E., The in-line cryotron. *Proc. I.R.E.* **51**:442 , 1963.
- 8a Brooks, F. P. Mass memory in computer systems. *I.E.E.E. Trans. on Magnetics* **MAG 5**: 635, 1969.
9. Conti, C.J. Concepts for buffer storage. *I.E.E.E. Computer Group News*, **2**: March, 9, 1969.
10. Cooley, J.W., and Tukey, J.W., An algorithm for the machine calculation of complex Fourier series. *Math. and Computation.* **19**:297, 1965.
11. Cooper, F.S., Liberman, A.M., and Borst, J.M., The inter-conversion of audible and visible patterns as a basis for research in the perception of speech. *Proc. Nat. Acad. Sci.*, **37**:318, 1951.
12. Dewey, G., *Relative Frequency of English Speech Sounds*. Harvard Univ. Press, Cambridge, Mass., 1923.
13. Edwards, H.H., and Newhouse, V.L., The crossed-film cryotron and its applications. in *Amplifier and Memory Devices with Films and Diodes*. N.S. Prywer, ed., McGraw-Hill, New York, 1965, pp. 391-409.
14. Elias, P., Information theory. in *Handbook of Automation, Computation and Control*, **Vol. I**. E.M. Grabbe, S. Ramo, and D.E. Wooldridge, eds., Wiley, New York, 1958, pp. 16-01 - 16-48.
15. Fano, R.M., *Transmission of Information. A Statistical Theory of Communication*. Wiley, New York, 1961.
16. Fant, G., *Vocoder system*. U.S. Patent **3,346,695**, 1967.
17. Flanagan, J.L., *Speech Analysis, Synthesis and Perception*. Academic Press, New York, 1965.
18. Friedman, A.P., and Rubin, J., On some invariant criteria for grouping data. *J.Am. Statistical Ass.* **62**: 1159, 1967.
19. Gates, J.F., Random-access mass memory system. in *Information Processing 1965*, Proc. of I.F.I.P. Congress 65, **Vol.II**, W.A. Kalenich, ed., Spartan, Washington, 1966, p. 596.
20. Gerstman, L.J., Classification of self-normalized vowels. *I.E.E.E. Trans. Audio and Electroacoustics*, **AU-16**: 78, 1968.

21. Gibson, D.H., and Shevel, W.L., "Cache" turns up a treasure. *Electronics*, **42**, Oct. 13, 105, 1969.
22. Haggard, M.P., and Mattingly, I.P., A simple program for synthesizing British English. *I.E.E.E. Trans. Audio and Electroacoustics*, **AU-16**: 95, 1968.
23. Hanna, P.R., Hanna, J.S., Hodges, R.E., and Rudolf, E.H., *Phoneme- Grapheme Correspondences as Cues to Spelling Improvement*. U.S. Govt. Printing Office, Washington, 1966.
24. Hanson, W.H., Ternary threshold logic. *I.R.E. Trans. on Electronic Computers*, **EC-11**: 191, 1963.
25. Ho, Y., and Agrawala, A.V., On pattern classification algorithms: Introduction and survey. *Proc. I.E.E.E.* **56**: 2101, 1968.
26. Huffman, D.A., A method for the construction of minimal redundancy codes. *Proc. I.R.E.* **40**: 1098, 1952.
- 26a. Huttenhoff, J. H., and Shively, R. R. Arithmetic unit of a computing element in a global, highly parallel computer. *I.E.E.E. Trans. on Comp.* **EC 18**: 695, 1969.
27. Jardine, N., and Sibson, R., Construction of hierarchic and non-hierarchic classifications. *Computer J.* **11**: 177, 1968.
28. Kelly, J.M., and Kennedy J R.N., Experimental cepstrum pitch detector for use in a 2400 bit/sec. channel vocoder. *J. Acous. Soc.* **40**: 1241, 1966.
29. King, G.W., Mass-memory centered processing. in *Information Processing 1965*, Proceedings of I.F.I.P. Congress 65, **Vol. II**, W.A. Kalenich, ed., Spartan, Washington, 1966, pp. 596-597.
30. Kohlenberg, A., 9600 BPS-A magic speed for data transmission. *Telecom.* **2**: Nov., 13, 1968.
31. Lance, G.N., and Williams, W.T., A general theory of classificatory sorting. I. Hierarchical systems. *Computer J.* **9**: 373, 1967.
- 31a. Latter, R.F., Microwave radio and coaxial cable facilities in the long distance telephone network. *Telecommunications* **2**: Feb., 11, 1969.
32. Laura, J., and Eng, A., Now, a trillion-bit computer mass memory. *Sci. Res.* July 21, 1969.
33. Lefkovitz, D., *File Structures for On-Line Systems*, Spartan, New York, 1969.
34. Liberman, A.M., Delattre, P.C., Cooper, F.S., and Gerstman, L.J., The role of consonant-vowel transitions in the stop and nasal consonants. *Psychol. Monographs* **68**, **No. 379**, 1954.
35. Lindquist, A.B., Content addressable memories; panel report. in *Information Processing 1965*, Proc. of I.F.I.P. Congress 65, **Vol. II**, W.A. Kalenich, ed., Spartan, Washington, 1966, p.479.

36. Lyght, C.E., ed. *The Merck Manual of Diagnosis and Therapy*. Merck & Co., Rahway, N.J., 1966.
37. Manheim, M.L., *Hierarchical Structure: A Model of Design and Planning Processes*. M.I.T. Press, Cambridge, Mass., 1966.
38. Mattingly, I.P., Experimental methods for speech synthesis by rule. *I.E.E.E. Trans. Audio and Electroacoustics*, **AU-16**: 198, 1968.
39. Mayne, J.G., Weksel, W., and Shola, P.N., Toward automating medical history. *Mayo Clin. Proc.* **43**: 1, 1968.
- 39a. Mayr, E. *Principles of Systematic Zoology*. McGraw-Hill, New York, 1969.
40. McQuitty, L.L., and Clark, J.A., Clusters from iterative intercolumn correlation analysis. *Educ. Psych. Meas.* **28**: 211, 1968.
41. Miller, G.A., Speech and language. in *Handbook of Experimental Psychology*, S.S. Stevens, ed., Wiley, New York, 1951, pp. 789-810.
42. Miller, R.L., Vocoder system for commercial telephone speech. *J. Acoust. Soc.* **40**: 1241, 1966.
- 42a. Minsky, M. *Semantic Information Processing*. M.I.T. Press, Cambridge, Mass., 1968.
43. Minsky, M., and Papert, S., *Perceptrons*. M.I.T. Press, Cambridge, Mass., 1969.
44. Nagy, G., State of the art in pattern recognition. *Proc. I.E.E.E.* **56**: 836, 1968.
45. Noll, A.M., Short-time spectrum and 'cepstrum' techniques for vocal pitch detection. *J. Acoust. Soc. Am.* **36**: 296, 1964.
46. Noll, A.M., Cepstrum pitch determination. *J. Acoust. Soc. Am.* **41**: 293, 1967.
47. Oettinger, A.G., and King, G.W., Letters to the editor on "Machine Translation of Chinese" *Sci. Amer.* **209**: Oct., 8, 1963.
- 47a. Olson, H. F., and Belar, H. Syllable analyzer, coder and synthesizer for transmission of speech. *I.R.E. Trans. on Audio.* **AU 10**: 11, 1962.
- 47b. Olson, H. F., Belar, H., and Rodgers, E. S. Research towards 8 high efficiency voice communications system. *J. Audio Eng. Soc.* **14**: 233, 1966.
48. Oppenheim, A.V., Speech analysis-synthesis system based on homomorphic filtering. *J. Acoust. Soc.* **45**: 458, 1969.
49. Ornstein, L., Computer learning and the scientific method: a proposed solution to the information theoretical problem of meaning. *J. Mt. Sinai Hosp., N.Y.C.* **32**: 437, 1965.
- 49a. Ornstein, L., *Data Processing*, U. S. Patent **3,633,171**, 1972
50. Pantin, C.F.A., *The Relations Between the Sciences*. Cambridge Univ. Press, New York, 1968.

51. Peterson, W.W., Error-correcting codes, *Sci. Amer.*, **206**: Feb. ,96, 1962 .
52. Pierce, J.R., The transmission of computer data . *Sci . Amer.* **215** : Sept., 144, 1966.
53. Pols, L.C.W., van der Kamp, L.J. Th., and Plomp, R., Perceptual and physical space of vowel sounds . *J. Acous . Soc . Am.* **46** : 458, 1969 .
54. Rabiner, L.R., Levitt, H., and Rosenberg, A.E., Investigation of stress patterns for speech synthesis by rule. *J. Acoust. Soc.* **45** : 92, 1969 .
55. Raisbeck, G., *Information Theory*. M.I.T. Press, Cambridge, Mass.,
56. Reddy, D.R., Computer recognition of connected speech. *J. Acoust. Soc.* **42** : 329, 1967 .
- 56a. Reitboeck, H., and Brody, T. P. Transformation with invariance under cyclic permutation for applications in pattern recognition. *Inform, Control* **15**: 130, 1969.
- 56b. Rosenfeld, J.L.\_A case study in programming for parallel processors. *Comm. ACM* **12**: 645, 1969.
- 56c. Ross, P. W. A limited-vocabulary adaptive speech-recognition system. *J. Audio Eng. Soc.* **15**: 414, 1967.
57. Scarrott, G.G., The efficient use of multilevel storage. in *Information Processing 1965*, Proc . of I .F. I .P. Congress 65, Vol . II \_ W.A. Kalenich, ed., Spartan, Washington, 1966, p.479.
58. Schroeder, M.R., Similarity measure for automatic speech and speaker recognition. *J. Acoust . Soc .* **43** : 375, 1968 .
59. Shannon, C.E., and Weaver, W., *The Mathematical Theory of Communication*. Univ. Ill. Press, Urbana, 1949.
60. Shepard , R. N., The analysis of proximities : multidimensional scaling with an unknown distance function. *Psychometrika* , **27** : 125 and 219, 1962 .
61. Shepard, R.N., Metric structures in ordinal data. *Behavior Sci.* **9**: 57, 1964 .
62. Shugart, A.F., and Tang, Y., IBM 2321 Data Cell Drive. *A.F.I.P.S. Conference Proceedings* . **28** : 335, 1966 .
63. Smith, F.D., Error considerations in data transmission. *Data Processing Mag* **11** : Nov., 32 , 1969 .
64. Smits , F .M., and Gallaher , L . E ., Design considerations for a semipermanent optical memory. *Bell Sys . Tech. J.* **46** : 1267, 1967 .
65. Weiss, M.R., Vogel, R.P., and Harris, C.M., Implementation of a pitch extractor of the double-spectrum-analysis type. *J. Acoust. Soc.* **40** : 657, 1966 .
66. White, W.L., Erickson, M.M., and Stevens, S.C., *Practical Automation for the Clinical Laboratory* . C .V . Mosby Co ., St . Louis , 1968 .

67. Whittaker, R.H., New concepts of Kingdoms of organisms. *Science*, **163**: 150, 1969
68. Wright, A.H., and Wright, A.A., *Handbook of Frogs and Toads of the United States and Canada*. Comstock, Ithaca, N.Y., 1949.
69. Yngve, V.H., Implications of mechanical translation research. *Proc. Am. Phil. Soc.* **108**: 275, 1964.
70. Zadeh, L.A., Fuzzy sets. *Inf. Control*, **8**: 338, 1965.
71. Zipf, G., *The Psycho-Biology of Language*. M.I.T. Press, Cambridge, Mass., 1965.