Finitary Models of Language Users

In this chapter we consider some of the models and measures that have been proposed to describe talkers and listeners—to describe the users of language rather than the language itself. As was pointed out at the beginning of Chapter 12, our language is not merely the collection of our linguistic responses, habits, or dispositions, just as our knowledge of arithmetic is not merely the collection of our arithmetic responses, habits, or dispositions. We must respect this distinction between the person's knowledge and his actual or even potential behavior; a formal characterization of some language is not simultaneously a model of the users of that language.

When we turn to the description of a user, a severe constraint is placed on our formulations. We have seen that natural languages are not adequately characterized by one-sided linear grammars (finite automata), yet we know that they must be spoken and heard by devices with bounded memory. How might this be accomplished? No automaton with bounded memory can produce all and only the grammatical sentences of a natural language; every such device, man presumably included, will exhibit certain limitations.

In considering models for the actual performance of human talkers and listeners an important criterion of adequacy and validity must be the extent to which the model's limitations correspond to our human limitations. We shall consider various finite systems—both stochastic and algebraic—with the idea of comparing their shortcomings with those of human talkers and listeners. For example, the fact that people are able to produce and comprehend an unlimited variety of novel sentences indicates immediately that their capacities are quite different from those of an automaton that compiles a simple list of all the grammatical sentences it hears. This example is trivial, yet it illustrates the kind of argument we must be prepared to make.

1. STOCHASTIC MODELS

It is often assumed, usually by workers interested in only one aspect of communication, that our perceptual models for a listener will be rather different from any behavioral models we might need for a speaker.

That assumption was not adopted in our discussion of formal aspects of linguistic competence, and it will not be adopted here in discussing empirical aspects of linguistic performance. In proposing models for a *user* of language—a user who is simultaneously talker and listener—we have assumed instead that the theoretically significant aspects of verbal behavior must be common to both the productive and receptive functions.

Once a formal theory of communication or language has been constructed, it generally turns out to be equally useful for describing both sources and receivers; in order to describe one or the other we simply rename various components of the formal theory in an appropriate fashion. This is illustrated by the stochastic theories considered in this section.

Stochastic theories of communication generally assume that the array of message elements can be represented by a probability distribution and that various communication processes (coding, transmitting, and receiving) have the effect of operating on that a priori distribution to transform it according to known transitional probabilities into an a posteriori distribution. The basic mathematical idea, therefore, is simply the multiplication of a vector by a matrix. But the interpretation we give to this underlying mathematical structure differs, depending on whether we interpret it as a model of a source, a channel, or a receiver. Thus the distinction between talkers and listeners is in no way critical for the development of the basic stochastic theory of communication. The same neutrality also characterizes the algebraic models of the user that are discussed in Sec. 2 of this chapter.

Purely for expository purposes, however, it is often convenient to present the mathematical argument in a definite context. For that reason we have arbitrarily chosen here to interpret the mathematics as a model of the source. This choice should not be taken to mean that a stochastic theory of communication must be concerned solely, or even principally, with speakers rather than with transmitters or hearers. The parallel development of these models for a receiver would be simply redundant, since little more than a substitution of terms would be involved.

1.1 Markov Sources

An important function of much communication is to reduce the uncertainty of a receiver about the state of affairs existing at the source. In such task-oriented communications, if there were no uncertainty about what a talker would say, there would be no need for him to speak. From a receiver's point of view the source is unpredictable; it would seem to

be a natural strategy, therefore, to describe the source in terms of probabilities. Moreover, the process of transmission is often exposed to random and unpredictable perturbations that can best be described probabilistically. The receiver himself is not above making errors; his mistakes can be a further source of randomness. Thus there are several valid motives for the development of stochastic theories of communication.

A stochastic theory of communication readily accommodates an infinitude of alternative sentences. Indeed, there would seem to be far more stochastic sequences than we actually need. Since no grammatical sentence is infinitely long, there can be at most only a countable infinitude of them. In probability theory we deal with a random sequence that extends infinitely in both directions, past and future, and we consider the uncountable infinitude of all such sequences that might occur.[2] The events with which probability theory deals are subsets of this set of all sequences. A finite stochastic sentence, therefore, must correspond to a finite segment of the infinite random sequence. A probability measure is assigned to the space of all possible sequences in such a way that (in theory, at least) the probability of any finite segment can be computed.

If the process of manufacturing messages were completely random, the product would bear little resemblance to actual utterances in a natural language. An important feature of a stochastic model for verbal behavior is that successive symbols can be correlated—that the history of the message will support some prediction about its future. In 1948 Shannon revived and elaborated an early suggestion by Markov that the source of messages in a discrete communication system could be represented by a stationary stochastic process that selected successive elements of the message from a finite vocabulary according to fixed probabilities. For example, Markov (1913) classified 20,000 successive letters in Puskhin's *Eugene Onegin* as vowels v or consonants c, then tabulated the frequency N of occurrences of overlapping sequences of length three. His results are summarized in Table 1 in the form of a tree.

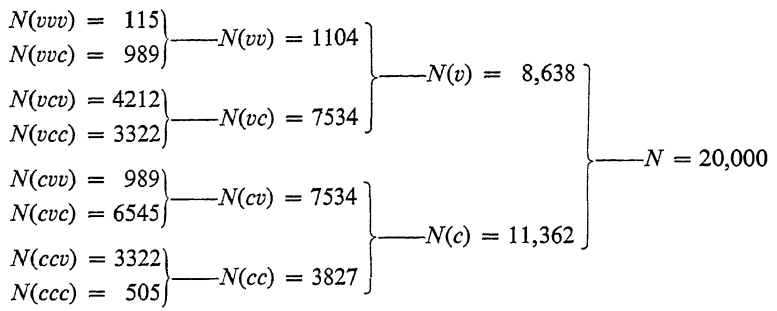
There are several constraints on the frequencies that can appear in such a tabulation of binary sequences. For example, $N(vc) = N(cv) \pm 1$, since the sequence cannot shift from vowels to consonants more often, ± 1, than it returns from consonants to vowels. In this particular example the number of degrees of freedom is 2^{n-1}, where n is the length of the string that is analyzed and 2 is the size of the alphabet.

The tabulated frequencies enable us to estimate probabilities. For instance, the estimated probability of a vowel is $\hat{p}(v) = N(v)/N = 0.432$. If successive letters were independent, we would expect the probability of a vowel following a consonant $p(v \mid c)$ to be the same as the probability of a

[2] We assume that the stochastic processes we are studying are stationary.

vowel following another vowel $p(v \mid v)$, and both would equal $p(v)$. The tabulation, however, yields $\hat{p}(v \mid c) = 0.663$, which is much larger than $\hat{p}(v)$, and $\hat{p}(v \mid v) = 0.128$, which is much smaller. Clearly, Russian vowels are more likely to occur after consonants than after vowels. Newman (1951) has reported further data on the written form of several languages and has confirmed this general tendency for vowels and consonants to alternate. (It is unlikely that this result would be seriously affected if the analyses had been made with phonemes rather than with written characters.)

Table 1 Markov's Data on Consonant-Vowel Sequences in Pushkin's *Eugene Onegin*



Inspection of the message statistics in Table 1 reveals that the probability of a vowel depends on more than the one immediately preceding letter. Strictly speaking, therefore, the chain is not Markovian, since a Markov process has been defined in such a way (cf. Feller, 1957) that all of the relevant information about the history of the sequence is given when the single, immediately preceding outcome is known. However, the Markovian representation is readily projected to handle more complicated cases. We shall consider how this can be done.

But first we must clarify what is meant by a Markov source. Given a discrete Markov process with a finite number of states $v_0, \ldots, v_D$ and a probability measure μ, a *Markov source* is constructed by defining $V = \{v_0, \ldots, v_D\}$ to be the vocabulary; messages are formed by concatenating the names of the successive states through which the system passes. In the terms used in Sec. 1.2 of Chapter 12 a Markov source is a special type of finite state automaton in which the triples that define it are all of the form (i, j, i) and in which the control unit has access to the conditional probabilities of all state transitions.

In Sec. 2 of Chapter 11, a state was defined as the set of all initial strings that were equivalent on the right. This definition must be extended for stochastic systems, however. We say that all the strings that allow the same

continuations with the same probabilities are stochastically equivalent on the right; then a state of a stochastic source is the set of all strings that are stochastically equivalent on the right.

If we are given a long but arbitrary sequence of symbols and wish to test whether it comprises a Markov chain, we must proceed to tabulate the frequencies of the possible pairs, triplets, etc. Our initial (Markovian) hypothesis in this analysis is that the symbol occurring at any given time can be regarded as the name of the state that the source is in at that time. Inspection of the actual sequence, however, may reveal that some of the hypothesized states are stochastically equivalent on the right (all possible continuations are assigned the same probabilities in both cases) and so can be parsimoniously combined into a single state. This reduction in the number of states implies that the state names must be distinguished from the terminal vocabulary. We can easily broaden our definition of a Markov source to include these simplified versions by distinguishing the set of possible states $\{S_0, S_1, \ldots, S_m\}$ from the vocabulary $\{v_0, v_1, \ldots, v_D\}$.

Since human messages have dependencies extending over long strings of symbols, we know that any pure Markov source must be too simple for our purposes. In order to generalize the Markov concept still further, therefore, we can introduce the following construction (McMillan, 1953): given a Markov source with a vocabulary V, select some different vocabulary W and define a homomorphic mapping of V into W. This mapping will define a new probability measure. The new system is a *projection* of a Markov source, but it may not itself be Markovian in the strict sense.

Definition 1. *Given a Markov source with vocabulary* $V = \{v_0, \ldots, v_D\}$, *with internal states* $S_0, \ldots, S_m$, *and with probability measure* μ, *a new source can be constructed with the same states but with vocabulary* W *and derived probability measure* μ', *where* $w_j \in W$ *if and only if there is a* $v_i \in V$, *and a mapping* θ *such that* $\theta(v_i) = w_j$. *Any source formed from a Markov source by this construction is a* projected Markov source.

The effect of this construction is best displayed by an example. Consider the Markov source whose graph is shown in Fig. 1 and assume that appropriate probabilities are assigned to the indicated transitions, all other conceivable transitions having probability zero. The vocabulary is $V = \{1, 2, 3, 4\}$, and each symbol names the state that the system is in after that symbol occurs. We shall consider three different ways to map V into an alternative vocabulary according to the construction in Definition 1:

1. Let $\theta(1) = \theta(4) = v$ and $\theta(2) = \theta(3) = c$. Then the projected system is a *higher order Markov source* of the type required to represent the probabilities of consonant-vowel triplets in Table 1. Under this construction we would probably identify state 1 as [*vv*], state 2 as [*vc*], state 3

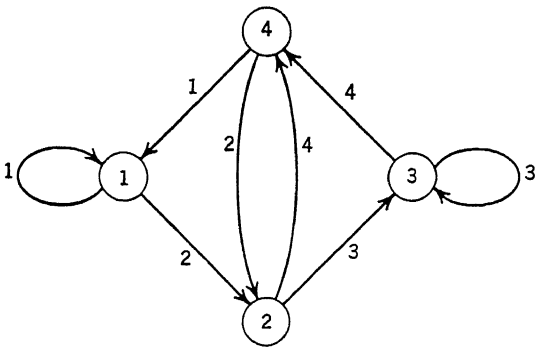


Fig. 1. Graph of a Markov source.

as [*cc*], and state 4 as [*cv*], thus retaining the convention of naming states after the sequences that lead into them, but now with the non-Markovian stipulation that more than one preceding symbol is implicated. In the terminology of Chapter 12, we are dealing here with a k-limited automaton, where $k = 2$.

2. Let $\theta(1) = \theta(2) = a$ and $\theta(3) = \theta(4) = b$. Then the projected system is ambiguous: an occurrence of a may leave the system in either state 1 or state 2; an occurrence of b may leave it in either state 3 or state 4. The states cannot be distinctively named after the sequences that lead into them.

3. Let $\theta(1) = -1$, $\theta(2) = \theta(4) = 0$, and $\theta(3) = +1$. With this projection we have a non-Markovian example mentioned by Feller (1957, p. 379). If we are given a sequence of independent random variables that can assume the values ± 1 with probability $\frac{1}{2}$, we can define the moving average of successive pairs, $X_n = (Y_n + Y_{n+1})/2$. The sequence of values of X_n is non-Markovian for an instructive reason; given a consecutive run of $X_n = 0$, how it will end depends on whether it contains an odd or an even number of 0's. After a run of an even number of occurrences of $X_n = 0$ the run must terminate as it began; after an odd number the run must terminate with the opposite symbol from the one with which it started. Thus it is necessary to remember how the system got into each run of 0's and how long the run has been going on. But, since there is no limit to how long a run of 0's may be, this system is not k-limited for any k. Thus it is impossible to produce the moving average by a simple Markov source or even by a higher order (k-limited) Markov process (which still must have finite memory), but it is quite simple to produce it with the projected Markov source constructed here.

By this construction, therefore, we can generalize the notion of a Markov

source to cover any kind of finite state system (regular event) for which a suitable probability measure has been defined.

Theorem 1. *Any finite state automaton over which an appropriate probability measure is defined can serve as a projected Markov source.*

Given any finite state automaton with an associated probability measure, assign a separate integer to each transition. The set of integers so assigned must form the vocabulary of a Markov source, and the rule of assignment defines a homomorphic mapping into a projected Markov source. This formulation makes precise the sense in which regular languages can be said to have Markovian properties.

All of our projected Markov sources will be assumed to operate in real time, from past to future, which we conventionally denote as left to right. Considered as rewriting systems, therefore, they contain only right-branching rules of the general form $A \rightarrow aB$, where A and B correspond to states of the stochastic system. The variety of projected Markov sources is, of course, extremely large, and only a few of the many possible types have been studied in any detail. We shall sample some of them in the following sections.

These same ideas could have been developed equally well to describe a receiver rather than a source. A projected Markov receiver is one that will accept as input only those strings of symbols that correspond to possible sequences of state transitions and that, through explicit agreement with the source or through long experience, has built up for each state an estimate of the probabilities of all possible continuations. As we have already noted, once the mathematical theory is fixed its specialization as a model for either the speaker or the hearer is quite simple. We are really concerned with ways to characterize the user of natural languages; the fact that we have here pictured him as a source is quite arbitrary.

1.2 k-Limited Stochastic Sources

One well-studied type of projected Markov source is known generally as a higher order, or k-limited, Markov source, which generates a $(k + 1)$-order approximation to the sample of text from which it is derived. The states of the k-limited automaton are identified with the sequences of k successive symbols leading into them, and associated with each state is a probability distribution defined over the D different symbols of the alphabet. If there are D symbols in the alphabet, then a k-limited stochastic source will have (potentially) D^k different states. A 0-limited stochastic source has but one state and generates the symbols independently.

If k is small and if we consider an alphabet of only 27 characters (26

letters and a space), it is possible to estimate the transitional probabilities for a k-limited stochastic source by actually counting the number of $(k + 1)$-tuplets of each type in a long sample of text. If we use these tabulations, it is then possible to produce $(k + 1)$-order approximations to the original text by drawing successive characters according to the probability distribution associated with the state determined by the string of k preceding characters. It is convenient to define a zero-order approximation as one that uses the characters independently and equiprobably; a first-order approximation uses the characters independently; a second-order approximation uses the characters with the probabilities appropriate in the context of the immediately preceding letter; etc.

An impression of the kind of approximations to English that these sources produce can be obtained from the following examples, taken from Shannon (1948). In each case the $(k + 1)$th symbol was selected with probability appropriate to the context provided by the preceding k symbols.

1. Zero-order letter approximation (26 letters and a space, independent and equiprobable): XFOML RXKHRJFFJUJ ZLPWCFWKCYJ FFJEYVKCQSGHYD QPAAMKBZAACIBZLHJQD.

2. First-order letter approximation (characters independent but with frequencies representative of English): OCRO HLI RGWR NMIELWIS EU LL NBNESEBYA TH EEI ALHENHTTPA OOBTTVA NAH BRL.

3. Second-order letter approximation (successive pairs of characters have frequencies representative of English text): ON IE ANTSOUTINYS ARE T INCTORE ST BE S DEAMY ACHIN D ILONASIVE TUCOOWE AT TEASONARE FUSO TIZIN ANDY. TOBE SEACE CTISBE.

4. Third-order letter approximation (triplets have frequencies representative of English text): IN NO IST LAT WHEY CRATICT FROURE BIRS GROCID PONDENOME OF DEMONSTURES OF THE REPTAGIN IS REGOACTIONA OF CRE.

A k-limited stochastic source can also be defined for the words in the vocabulary V in a manner completely analogous to that for letters of the alphabet A. When states are defined in terms of the k preceding words, the following kinds of approximations are obtained:

5. First-order word approximation (words independent, but with frequencies representative of English): REPRESENTING AND SPEEDILY IS AN GOOD APT OR CAME CAN DIFFERENT NATURAL HERE HE THE A IN CAME THE TO OF TO EXPERT GRAY COME TO FURNISHES THE LINE MESSAGE HAD BE THESE.

6. Second-order word approximation (word-pairs with frequencies

representative of English): THE HEAD AND IN FRONTAL ATTACK ON AN ENGLISH WRITER THAT THE CHARACTER OF THIS POINT IS THEREFORE ANOTHER METHOD FOR THE LETTERS THAT THE TIME OF WHO EVER TOLD THE PROBLEM FOR AN UNEXPECTED.

The following two illustrations are taken from Miller & Selfridge (1950).

7. Third-order word approximation (word-triplets with frequencies representative of English): FAMILY WAS LARGE DARK ANIMAL CAME ROARING DOWN THE MIDDLE OF MY FRIENDS LOVE BOOKS PASSIONATELY EVERY KISS IS FINE.

8. Fifth-order word approximation (word quintuplets with frequencies representative of English): ROAD IN THE COUNTRY WAS INSANE ESPECIALLY IN DREARY ROOMS WHERE THEY HAVE SOME BOOKS TO BUY FOR STUDYING GREEK.

Higher-order approximations to the statistical structure of English have been used to manipulate the apparent meaningfulness of letter and word sequences as a variable in psychological experiments. As k increases, the sequences of symbols take on a more familiar look and—although they remain nonsensical—the fact seems to be empirically established that they become easier to perceive and to remember correctly.

We know that the sequences produced by k-limited Markov sources cannot converge on the set of grammatical utterances as k increases because there are many grammatical sentences that are never uttered and so could not be represented in any estimation of transitional probabilities. A k-limited Markov source cannot serve as a natural grammar of English no matter how large k may be. Increasing k does not isolate the set of grammatical sentences, for, even though the number of high-probability grammatical sequences included is thereby increased, the number of low-probability grammatical sequences excluded is also increased correspondingly. Moreover, for any finite k there would be ungrammatical sequences longer than k symbols that a stochastic user could not reject.

Even though a k-limited source is not a grammar, it might still be proposed as a model of the user. Granted that the model cannot isolate the set of all grammatical sentences, neither can we; inasmuch as our human limitations often lead us into ungrammatical paths, the real test of this model of the user is whether it exhibits the same limitations that we do.

However, when we examine this model, not as a convenient way to summarize certain statistical parameters of message ensembles, but as a serious proposal for the way people create and interpret their communicative

utterances, it is all too easy to find objections. We shall mention only one, but one that seems particularly serious: the k-limited Markov source has far too many parameters (cf. Miller, Galanter, & Pribram, 1960, pp. 145–148). As we have noted, there can be as many as D^k probabilities to be estimated. By the time k grows large enough to give a reasonable fit to ordinary usage the number of parameters that must be estimated will have exploded; a staggering amount of text would have to be scanned and tabulated in order to make reliable estimates.

Just how large must k and D be in order to give a satisfactory model? Consider a perfectly ordinary sentence: *The people who called and wanted to rent your house when you go away next year are from California.* In this sentence there is a grammatical dependency extending from the second word (the plural subject *people*) to the seventeenth word (the plural verb *are*). In order to reflect this particular dependency, therefore, k must be at least 15 words. We have not attempted to explore how far k can be pushed and still appear to stay within the bounds of common usage, but the limit is surely greater than 15 words; and the vocabulary must have at least 1000 words. Taking these conservative values of k and D, therefore, we have $D^k = 10^{45}$ parameters to cope with, far more than we could estimate even with the fastest digital computers.

Of course, we can argue that many of these 10^{45} strings of 15 words whose probabilities must be estimated are redundant or that most of them have zero probability. A more realistic estimate, therefore, might assume that what we learn are not the admissible strings of words but rather the "sentence frames"—the admissible strings of syntactic categories. Moreover, we might recognize that not all sequences of categories are equally likely to occur; as a conservative estimate (cf. Somers, 1961), we might assume that on the average there would be about four alternative categories that might follow in any given context. By such arguments, therefore, we can reduce D to as little as four, so that D^k becomes $4^{15} = 10^9$. That value is, of course, a notable improvement over 10^{45} parameters, yet, when we recall that several occurrences of each string are required before we can obtain reliable estimates of the probabilities involved, it becomes apparent that we still have not avoided the central difficulty—an enormous amount of text would have to be scanned and tabulated in order to provide a satisfactory empirical basis for a model of this type.

The trouble is not merely that the statistician is inconvenienced by an estimation problem. A learner would face an equally difficult task. If we assume that a k-limited automaton must somehow arise during childhood, the amount of raw induction that would be required is almost inconceivable. We cannot seriously propose that a child learns the values of 10^9 parameters in a childhood lasting only 10^8 seconds.

1.3 A Measure of Selective Information

Although the direct estimation of all the probabilities involved in a k-limited Markov model of the language user is impractical, other statistics of a more general and summary nature are available to represent certain average characteristics of such a source. Two of these with particular interest for communication research are *amount of information* and *redundancy*. We introduce them briefly and heuristically at this point.

The problem of measuring amounts of information in a communication situation seems to have been posed first by Hartley (1928). If some particular piece of equipment—a switch, say, or a relay—has D possible positions, or physical states, then two of the devices working together can have D^2 states, three can have D^3 states altogether, etc. The number of possible states of the total system increases exponentially as the number of devices increases linearly. In order to have a measure of information that will make the capacity of $2n$ devices just double the capacity of n of them, Hartley defined what we now call the information capacity of a device as $\log D$, where D is the number of different states the total system can get into. Hartley's proposal was later generalized and considerably extended by Shannon (1948) and Wiener (1948).

When applied to a communication channel, Hartley's notion of capacity refers to the number of different signals that might be transmitted in a unit interval of time. For example, let $N(T)$ denote the total number of different strings exactly T symbols long that the channel can transmit. Let D be the number of different states the channel has available and assume that there are no constraints on the possible transitions from one state to another. Then $N(T) = D^T$, or

$$\frac{\log N(T)}{T} = \log D,$$

which is Hartley's measure of capacity. In case there are some constraints on the possible transitions, $N(T)$ will still (in the limit) increase exponentially but less rapidly. In the general case, therefore, we are led to define channel capacity in terms of the limit:

$$\text{channel capacity} = \lim_{T \to \infty} \frac{\log N(T)}{T}. \tag{1}$$

This is the best the channel can do. If a source produces more information per symbol on the average, the channel will not be able to transmit it all—not, at least, in the same number of symbols. The practical problem,

therefore, is to estimate $N(T)$ from what we know about the properties of the channel.

Our present goal, however, is to see how Hartley's original insight has been extended to provide a measure of the amount of information per symbol contained in messages generated by stochastic devices of the sort described in the preceding sections of this chapter. We shall confine our discussion here to those situations in which the purpose of communication is to reduce a receiver's uncertainty. The amount of information he receives, therefore, must be some function of what he learns about the state of the source. And what he learns will depend on how ignorant he was to begin with. Let us assume that the source selects its message by any procedure, random or deterministic, but that all the receiver knows in advance is that the source will choose among a finite set of mutually exclusive messages $M_1, M_2, \ldots, M_D$ with probabilities $p(M_1), p(M_2), \ldots, p(M_D)$, where these probabilities sum to unity. What Shannon and Wiener did was to develop a measure $H(M)$ of the receiver's uncertainty, where the argument M designates the choice situation:

$$M = \begin{pmatrix} M_1, & M_2, & \ldots, & M_D \\ p(M_1), & p(M_2), & \ldots, & p(M_D) \end{pmatrix}.$$

When the particular message is correctly received, a listener's uncertainty about it will be reduced from $H(M)$ to zero; therefore, the message conveyed $H(M)$ units of information. Thus $H(M)$ is a measure of the amount of information required (on the average) to select M_i when faced with the choice situation M.

We list as assumptions a number of properties that intuition says a reasonable measure of uncertainty ought to have for discrete devices. Then, following a heuristic presentation by Khinchin (1957), we shall use those assumptions to develop the particular H of Shannon and Wiener.

Our first intuitive proposition is that uncertainty depends only on what might happen. Impossible events will not affect our uncertainty. If a particular message M_i is known in advance to have $p(M_i) = 0$, it should not affect the measure H in any way if M_i is omitted from consideration.

Assumption 1. *Adding any number of impossible messages to M does not change H(M):*

$$H\begin{pmatrix} M_1, & \ldots, & M_D, & M_{D+1} \\ p(M_1), & \ldots, & p(M_D), & 0 \end{pmatrix} = H\begin{pmatrix} M_1, & \ldots, & M_D \\ p(M_1), & \ldots, & p(M_D) \end{pmatrix}.$$

Our second intuition is that people are most uncertain when the alternative messages are all equally probable. Any bias that makes one message

more probable than another conveys information in the sense that it reduces the receiver's total amount of uncertainty. With only two alternative messages, for example, a 50 : 50 split presents the least predictable situation imaginable. Since there are D different messages in M, when they are equiprobable $p(M_i) = 1/D$ for all i.

Assumption 2. *$H(M)$ is a maximum when all the messages in M are equiprobable:*

$$H\begin{pmatrix} M_1, & \ldots, & M_D \\ p(M_1), & \ldots, & p(M_D) \end{pmatrix} \leqslant H\begin{pmatrix} M_1, & \ldots, & M_D \\ 1/D, & \ldots, & 1/D \end{pmatrix}.$$

Now let $L(D)$ represent the amount of uncertainty involved when all the messages are equiprobable. Then we have, by virtue of our two assumptions,

$$\begin{aligned} L(D) &= H\begin{pmatrix} M_1, & \ldots, & M_D, & M_{D+1} \\ 1/D, & \ldots, & 1/D, & 0 \end{pmatrix} \\ &\leqslant H\begin{pmatrix} M_1, & \ldots, & M_{D+1} \\ 1/D+1, & \ldots, & 1/D+1 \end{pmatrix} = L(D+1). \end{aligned}$$

Therefore, we have established the following lemma:

Lemma 1. *$L(D)$ is a monotonic increasing function of D.*

That is to say, when all D of the alternative messages in M are equiprobable $H(M)$ is a nondecreasing function of D. Intuitively, the more different things that can happen, the more uncertain we are.

It is also reasonable to insist that the uncertainty associated with a choice should not be affected by making the choice in two or more steps, but should be the weighted sum of the uncertainties involved in each step. This critically important assumption can be stated:

Assumption 3. *$H(M)$ is additive.*

Let any two events of M be combined to form a single, compound event, which we designate as $M_1 \cup M_2$ and which has probability $p(M_1 \cup M_2) = p(M_1) + p(M_2)$. Thus we can decompose M into two parts:

$$M' = \begin{pmatrix} M_1 \cup M_2, & M_3, & \ldots, & M_D \\ p(M_1) + p(M_2), & p(M_3), & \ldots, & p(M_D) \end{pmatrix},$$

and

$$M'' = \begin{pmatrix} M_1, & M_2, & M_3, & \ldots, & M_D \\ \dfrac{p(M_1)}{p(M_1)+p(M_2)}, & \dfrac{p(M_2)}{p(M_1)+p(M_2)}, & 0, & \ldots, & 0 \end{pmatrix}.$$

A choice from M is equivalent to a choice from M' followed (if $M_1 \cup M_2$ is chosen) by a choice from M''. Assumption 3 means that $H(M)$ depends

on the sum of $H(M')$ and $H(M'')$. In calculating $H(M)$, however, $H(M'')$ should be weighted by $p(M_1) + p(M_2)$ because that represents the probability that a second choice will be required. Assumption 3 implies that

$$H(M) = H(M') + [p(M_1) + p(M_2)]H(M'').$$

If this equation holds whenever two messages of M are lumped together, then it can easily be generalized to any subset whatsoever, and it can be extended to more than one subset of messages in M.

In order to discuss this more general situation, we represent the messages in M by M_{ij}, where i is the first selection and j is the second. The first selection is made from A:

$$A = \begin{pmatrix} A_1, & \ldots, & A_r \\ p(A_1), & \ldots, & p(A_r) \end{pmatrix},$$

where

$$p(A_i) = \sum_j p(M_{ij}),$$

and the second choice depends (as before) on the outcome of the first; that is to say, the second choice is made from

$$B \mid A_i = \begin{pmatrix} B_1, & \ldots, & B_s \\ p(B_1 \mid A_i), & \ldots, & p(B_s \mid A_i) \end{pmatrix}.$$

The B_j have probabilities $p(B_j \mid A_i)$ that depend on A_i, the preceding choice from A. The two choices together are equivalent to—are a decomposition of—a single choice from M, where

$$A_i B_j = M_{ij},$$

and

$$p(A_i)\, p(B_j \mid A_i) = p(M_{ij}).$$

Now, by Assumption 3, $H(M)$ should be the sum of the two components. But that is a bit complicated, since $H(B \mid A_i)$ is a random variable depending on i. On the average, however, it will be

$$E\{H(B \mid A_i)\} = \sum_i p(A_i)\, H(B \mid A_i) = H(B \mid A). \tag{2}$$

In this situation, therefore, the assumption of additivity means that

$$H(M) = H(AB) = H(A) + H(B \mid A). \tag{3}$$

Of course, if A and B are independent, Eq. 3 becomes

$$H(AB) = H(A) + H(B), \tag{4}$$

and, if the messages are independent and equally probable, a sequence of s successive choices among D alternatives will give

$$L(D^s) = sL(D). \tag{5}$$

We shall now establish the following lemma:

Lemma 2. $L(D) = k \log D$, *where* $k > 0$.

Consider repeated independent choices from the same number D of equiprobable messages. Select m such that for any positive integers D, s, t

$$D^m \leqslant s^t \leqslant D^{m+1} \tag{6}$$

$$m \log D \leqslant t \log s \leqslant (m+1) \log D$$

$$\frac{m}{t} \leqslant \frac{\log s}{\log D} \leqslant \frac{m+1}{t} \tag{7}$$

From Eq. 6, and the fact that $L(D)$ is monotonic increasing, it follows that

$$L(D^m) \leqslant L(s^t) \leqslant L(D^{m+1}),$$

and from Eq. 5 we know that

$$m L(D) \leqslant t L(s) \leqslant (m+1) L(D)$$

$$\frac{m}{t} \leqslant \frac{L(s)}{L(D)} \leqslant \frac{m+1}{t}. \tag{8}$$

Combining Eqs. 7 and 8, therefore,

$$\left| \frac{L(s)}{L(D)} - \frac{\log s}{\log D} \right| \leqslant \frac{1}{t}.$$

Since m is not involved, t can be chosen arbitrarily large, and

$$\frac{L(s)}{\log s} = \frac{L(D)}{\log D}.$$

Moreover, since D and s are quite arbitrary, these ratios must be constant independent of D; that is to say,

$$\frac{L(D)}{\log D} = k, \quad \text{so} \quad L(D) = k \log D.$$

Of course, $\log D$ is nonnegative and therefore [since $L(D)$ is monotonic increasing] $k > 0$. This completes the proof of Lemma 2.

Ordinarily k is chosen to be unity when logarithms are taken to the base 2,

$$L(D) = \log_2 D, \tag{9}$$

that is to say, the unit of measurement is taken to be the amount of uncertainty involved in a choice between two equally possible alternatives. This unit is called a *bit*.

Next consider the general case with unequal, but rational, probabilities. Let

$$p(A_i) = \frac{g_i}{g} \qquad (i = 1, \ldots, r),$$

where the g_i are all positive integers and

$$\sum_i g_i = g.$$

The problem is to determine $H(A)$. In order to do this, we shall construct a second choice situation $(B \mid A_i)$ in a special way so that the Cartesian product $M = A \times B$ will consist entirely of equiprobable alternatives.

Let $(B \mid A_i)$ consist of g_i messages each with probability $1/g_i$. Therefore,

$$H(B \mid A_i) = H\begin{pmatrix} B_1, & \ldots, & B_{g_i} \\ 1/g_i, & \ldots, & 1/g_i \end{pmatrix} = L(g_i) = c \log g_i. \tag{10}$$

From Eqs. 2 and 10 it follows that

$$\begin{aligned} H(B \mid A) &= \sum_i p(A_i)\, H(B \mid A_i) \\ &= \sum_i p(A_i) c \log g_i \\ &= c \sum_i p(A_i) \log p(A_i) g \\ &= c \log g + c \sum_i p(A_i) \log p(A_i)\,. \end{aligned} \tag{11}$$

Consider next the compound choice $M = A \times B$. Since

$$p(A_i B_j) = p(A_i)\, p(B_j \mid A_i) = \frac{g_i}{g} \cdot \frac{1}{g_i} = \frac{1}{g},$$

it must follow that for this specially contrived situation there are g equally probable events and

$$H(A \times B) = H(AB) = L(g) = c \log g. \tag{12}$$

When we substitute Eqs. 11 and 12 into Eq. 3 we obtain

$$c \log g = H(A) + c \log g + c \sum_i p(A_i) \log p(A_i).$$

We have now established the theorem:

Theorem 2. *For rational probabilities,*

$$H(A) = -c \sum_i p(A_i) \log p(A_i). \tag{13}$$

Since Eq. 13 can be interpreted as the mean value $E\{-\log p(A_i)\}$, the measure of uncertainty thus turns out to be the mean logarithmic probability—a quantity familiar to physicists under the name entropy. It is as

though we had defined the amount of information in message A_i to be $-\log p(A_i)$, regardless of what the probability distribution might be for the other messages. The assumption that the amount of information conveyed by one particular message is independent of all the other possible messages is what Luce (1960) has called the assumption of independence from irrelevant alternatives; he remarks (Luce, 1959) that it is characteristic—either explicitly or implicitly—of most theories of choice behavior.

Finally, in order to make $H(B)$ a continuous function of the probabilities, we need a fourth assumption of continuity. Since it is felt intuitively that a small change in probabilities should result in a small change in $H(M)$, this final assumption needs little comment here. It will not play a critical role in the discussion that follows.

Next, we want to use H to measure the uncertainty associated with the projected Markov sources. Suppose we have a stationary source with a finite number of states $A_1, \ldots, A_n$, with an alphabet $B_1, \ldots, B_D$, and with the matrix of transitional probabilities $p(B_j \mid A_i)$. When the system is in state A_i, the choice situation is

$$B \mid A_i = \begin{pmatrix} B_1, & B_2, & \ldots, & B_D \\ p(B_1 \mid A_i), & p(B_2 \mid A_i), & \ldots, & p(B_D \mid A_i) \end{pmatrix}.$$

By Theorem 2 the amount of information involved in this choice must be

$$H(B \mid A_i) = -c \sum_j p(B_j \mid A_i) \log p(B_j \mid A_i).$$

This quantity is defined for each state. In order to obtain an average value to represent the amount of information that we can expect for the source, regardless of the state it is in, we must average over i:

$$\begin{aligned} E\{H(B \mid A_i)\} &= \sum_i p(A_i) H(B \mid A_i) \\ &= -c \sum_i \sum_j p(A_i, B_j) \log p(B_j \mid A_i) = H(B \mid A). \end{aligned} \quad (14)$$

Now we can regard $H(B \mid A)$ as a measure of the average amount of information obtained when the source moves one step ahead by choosing a letter from the set $\{B_i\}$. [In the special case in which successive events in the chain are independent, of course, $H(B \mid A)$ reduces to $H(B)$.] A string of N successive choices, therefore, will yield $NH(B \mid A)$ units of information on the average.

In general, $H(AB) \leqslant H(A) + H(B)$; equality obtains only when A and B are independent. This fact can be demonstrated as follows: the familiar expansion

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots, \qquad (x \geqslant -1),$$

can be used to establish that $e^x \geqslant 1 + x$. If we set $t = 1 + x$, this inequality can be written as

$$t - 1 \geqslant \log_e t, \qquad (t \geqslant 0).$$

Now put $t = p(A_i)p(B_j)/p(A_iB_j)$:

$$\frac{p(A_i)p(B_j)}{p(A_iB_j)} - 1 \geqslant \log_e p(A_i) + \log_e p(B_j) - \log_e p(A_iB_j),$$

and take expected values over the distribution $p(A_iB_j)$:

$$\sum_j \sum_i p(A_iB_j) \frac{p(A_i)p(B_j)}{p(A_iB_j)} - 1 \geqslant \sum_j \sum_i p(A_iB_j) \log_e p(A_i) + \sum_j \sum_i p(A_iB_j) \log_e p(B_j) - \sum_j \sum_i p(A_iB_j) \log_e p(A_iB_j),$$

so

$$1 - 1 \geqslant -H(A) - H(B) + H(AB),$$

which is the result we wished to establish:

$$H(A) + H(B) \geqslant H(AB). \tag{15}$$

If we compare Eq. 15 with the assumption of additivity expressed in Eq. 3, we see that we have also established the following theorem:

Theorem 3. $$H(B) \geqslant H(B \mid A). \tag{16}$$

This important inequality can be interpreted to mean that knowledge of the choice from A cannot increase our average uncertainty about the choice from B. In particular, if A represents the past history of some message and B represents the choice of the next message unit, then the average amount of information conveyed by B can never increase when we know the context in which it is selected.

It is important to remember that H is an average measure of *selective* information, based on the assumption that the improbable event is always the most informative, and is not a simple measure of semantic information (cf. Carnap & Bar-Hillel, 1952). An illustration may suggest the kind of problems that can arise: in ordinary usage *It is a man* will generally be judged to convey more information than *It is a vertebrate*, because the fact that something is a man implies that it is a vertebrate, but not vice versa. In the framework of selective information theory, however, the situation is reversed. According to the tabulations of the frequencies of English words, *vertebrate* is a less probable word than *man*, and its selection in English discourse must therefore be considered to convey more information.

Because many psychological processes involve selective processes of one kind or another, a measure of selective information has proved to be of some value as a way to characterize this aspect of behavior. Surveys of various applications of information measures to psychology have been prepared by Attneave (1959), Cherry (1957), Garner (1962), Luce (1960), Miller (1953), Quastler (1955), and others. Not all applications of the mean logarithmic probability have been carefully considered and well motivated, however. As Cronbach (1955) has emphasized, in many situations it may be advisable to develop alternative measures of information based on intuitive postulates that are more closely related to the particular applications we intend to make.

1.4 Redundancy

Since $H(B) \geqslant H(B \mid A)$, where equality holds only for sequentially independent messages, any sequential dependencies that the source introduces will act to reduce the amount of selective information the message contains. The extent to which the information is reduced is a general and interesting property of the source. Shannon has termed it the *redundancy* and has defined it in the following way.

First, consider the amount of information that could be encoded in the given alphabet (or vocabulary) if every atomic symbol were used independently and equiprobably. If there are D atomic symbols, then the informational capacity of the alphabet will be $L(D) = \log_2 D$ bits per symbol. Moreover, this value will be the maximum possible with that alphabet. Now, if we determine that the source is producing an amount $H(M)$ that is actually less than its theoretical maximum per symbol, $H(M)/L(D)$ will be some fraction less than unity that will represent the *relative* amount of information from the source. One minus the relative information is the redundancy:

$$\text{per cent redundancy} = 100\left(1 - \frac{H(M)}{\log D}\right). \tag{17}$$

The relative amount of information per symbol is a measure of how efficiently the coding alphabet is being used. For example, if the relative information per symbol is only half what it might be, then on the average the messages are twice as long as necessary. Shannon (1948), on the basis of his observation that a highly skilled subject could reconstruct passages from which 50% of the letters had been removed, estimated the efficiency of normal English prose as something less than 50%. But, when Chapanis (1954) tried to repeat this observation with other subjects and other passages, he found that if letters are randomly deleted and the text is shortened

so that no indication is given of the location of the deletion few people are able to restore more than 25% of the missing letters in a short period of time. However, these are difficult conditions to impose on subjects. In order to estimate the coding efficiency of English writing, we should first make every effort to optimize the conditions for the person who is trying to reconstruct the text. For example, we might tell him in advance that all spaces between words and all vowels have been deleted. This form of abbreviation shortens the text by almost 50%, yet Miller and Friedman (1957) found that the most highly skilled subjects were able to restore the missing characters if they were given sufficient time and incentive to work at the task. We can conclude, therefore, that English is at least 50% redundant and perhaps more.

Why do we bother with such crude bounds? Why not compute redundancy directly from the message statistics for printed English? As we noted at the end of Sec. 1.2, the direct approach is quite impractical, for there are too many parameters to be estimated. However, we can put certain rough bounds on the value of H by limiting operations that use the available message statistics directly for short sequences of letters in English (Shannon, 1948). Let $p(x_i)$ denote the probability of a string x_i of k symbols from the source and define

$$G_k = -\frac{1}{k}\sum_i p(x_i)\log_2 p(x_i), \tag{18}$$

where the sum is taken over all strings x_i containing exactly k symbols. Then G_k will be a monotonic decreasing function of k and will approach H in the limit.

An even better estimate can be obtained with conditional probabilities. Consider a matrix P whose rows represent the D^k possible strings x_i of k symbols and whose columns represent the D different symbols a_j. The elements of the matrix are $p(a_j \mid x_i)$, the conditional probabilities that a_j will occur as the $(k+1)$st symbol given that the string x_i of k symbols just preceded it. For each row of this matrix

$$-\sum_j p(a_j \mid x_i)\log_2 p(a_j \mid x_i)$$

measures our uncertainty regarding what will follow the particular string x_i. The expected value of this uncertainty defines a new function,

$$F_{k+1} = -\sum_i \sum_j p(x_i)\, p(a_j \mid x_i)\log_2 p(a_j \mid x_i), \tag{19}$$

where $p(x_i)$ is the probability of string x_i. Since $p(x_i)\,p(a_j \mid x_i) = p(x_i a_j)$, we can show that

$$\begin{aligned} F_{k+1} &= (k+1)G_{k+1} - kG_k \\ &= (k+1)(G_{k+1} - G_k) + G_k. \end{aligned}$$

Therefore, as G_k approaches H, F_k must also approach H. Moreover,

$$G_{k+1} - F_{k+1} = \frac{k}{k+1} G_k \geqslant 0,$$

so we know that

$$G_k \geqslant F_k.$$

Thus F_k converges on H more rapidly than G_k as k increases.

Even F (and similar functions using the message statistics) converges quite slowly for natural languages, so Shannon (1951) proposed an estimation procedure using data obtained with a guessing procedure. We consider here only his procedure for determining an upper bound for H (and thus a lower bound for the redundancy).

Imagine that we have two identical k-limited automata that incorporate the true probabilities of English strings. Given a finite string of k symbols, these devices assign the correct probabilities for the $(k + 1)$st symbol. The first device is located at the source. As each symbol of the message is produced, the device guesses what the next symbol will be. It guesses first the most probable symbol, second the next most probable, and so on, continuing in this way until it guesses correctly. Instead of transmitting the symbol produced by the source, we transmit the number of guesses that the device required.

The second device is located at the receiver. When the number j is received, this second device interprets it to mean that the jth guess (given the preceding context) is correct. The two devices are identical and the order of their guesses in any context will be identical; the second machine decodes the received signal and recovers the original message. In that way the original message can be perfectly recovered, so the sequence of numbers must contain the same information—therefore no less an *amount* of information—as the original text. If we can determine the amount of information per symbol for the reduced text, we shall also have determined an upper bound for the original text.

What will the reduced text look like? We do not possess two such k-limited automata, but we can try to use native speakers of the language as a substitute. Native speakers do not know all the probabilities we need, but they do know the syntactic and semantic rules which lead to those probabilities. We can let a person know all of the text up to a given point, then on the basis of that and his knowledge of the language ask him to guess the next letter. Shannon (1951) gives the following as typical of the results obtained:

THERE#IS#NO#REVERSE#ON#A#...
1 1 1 5 1 1 2 1 1 2 1 1 1 5 1 1 7 1 1 1 2 1 3 2 1 2 2 ...

The top line is the original message; below it is the number of guesses required for each successive letter.

Note that most letters are guessed correctly on the first trial—approximately 80% when a large amount of antecedent context is provided. Note also that in the reduced text the sequential constraints are far less important; how many guesses the nth letter took tells little about how many will be needed for the $(n + 1)$st. It is as if the sequential redundancy of the original text were transformed into a nonsequential favoritism for small numbers in the reduced text. Thus we are led to consider the quantity

$$E_{k+1} = -\sum_{j=1}^{27} q_k(j) \log_2 q_k(j), \tag{20}$$

where $q_k(j)$ is the probability of guessing the $(k + 1)$st letter of a string correctly on exactly the jth guess. If k is large, and if our human subject is a satisfactory substitute for the k-limited automaton we lack, then E_k should be fairly close to H.

Can we make this idea more precise? Suppose we reconsider the $D^k \times D$ matrix P whose elements $p(a_j \mid x_i)$ are the conditional probabilities of sybmol a_j, given the string x_i. What the k-limited automaton will do when it guesses is to map a_j into the digit $\theta(a_j)$ for each row, where the character with the largest probability in the row would be coded as 1, the next largest as 2, and so on. Consider, therefore, a new $D^k \times D$ matrix Q whose rows are the same but whose columns represent the first D digits in order. Then in every row of this new matrix the conditional probabilities $q[\theta(a_j) \mid x_i]$ would be arranged in a monotonically decreasing order of magnitude from left to right. Note that we have lost nothing in shifting from P to Q; θ has an inverse, so F_k can be computed from Q just as well as from P.

Now suppose we ignore the context x_i; that is to say, suppose we simply average all the rows of Q together, weighting them according to their probability of occurrence. This procedure will yield $q_k(j)$, the average probability of being correct on the jth guess. From Theorem 3 we know that $F_k \leqslant E_k$. Therefore, E_k must also be an upper bound on the amount of information per symbol.

Moreover, this bound holds even when we use a human substitute for out hypothetical automaton, since people can err only in the direction of greater uncertainty (greater E_k) than would an ideal device. We can formulate this fact rigorously: suppose the true probabilities of the predicted symbols are p_i but that our subject is guessing on the basis of some (not necessarily accurate; cf. Toda, 1956) estimates $\hat{p}_i$, derived somehow from his knowledge of the language and his previous experience with the source. Let $\Sigma p_i = \Sigma \hat{p}_i = 1$, and consider the mean value of the

quantity $a_i = \hat{p}_i/p_i$. From the well-known theorem of the arithmetic and geometric means (see, e.g., Hardy, Littlewood, & Polya, 1952, Chapter 2), we know that

$$(a_1)^{p_1} \ldots (a_D)^{p_D} \leqslant p_1 a_1 + \ldots + p_D a_D,$$

from which we obtain directly

$$\left(\frac{\hat{p}_1}{p_1}\right)^{p_1} \ldots \left(\frac{\hat{p}_D}{p_D}\right)^{p_D} \leqslant 1,$$

with equality only if $\hat{p}_i = p_i$ for all i. Taking logarithms,

$$\sum_{i=1}^{D} p_i \log \frac{\hat{p}_i}{p_i} \leqslant 0,$$

which gives the desired inequality

$$-\sum p_i \log \hat{p}_i \geqslant -\sum p_i \log p_i. \tag{21}$$

Any inaccuracy in the subject's estimated probabilities can serve only to increase the estimate of the amount of information. The more ignorant he is, the more often the source will surprise him.

The guessing technique for estimating bounds on the amount of selective information contained in redundant strings of symbols can be performed rapidly, and the bounds are often surprisingly low. The technique has been useful even in nonlinguistic situations.

Shannon's (1951) data for a single, highly skilled subject gave $E_{100} = 1.3$ bits per character. For a 27-character alphabet the maximum possible would be $\log_2 27 = 4.73$ bits per character. The lower bound for the redundancy, therefore, is $1 - (1.3/4.73) = 0.73$. This can be interpreted to mean that, for the type of prose passages Shannon used, at least 73 of every 100 characters on the page could have been deleted if the same alphabet had been used most efficiently, that is, if all the characters were used independently and equiprobably. Burton and Licklider (1955) confirmed this result and added that E_k has effectively reached its asymptote by $k = 32$; that is to say, measurable effects of context on a person's guesses do not seem to extend more than 32 characters (about six words) back into the history of the message.

The lower bound on redundancy depends on the particular passage used. In some situations—air-traffic-control messages to a pilot landing at a familiar airport—redundancy may rise as high as 96% (Frick & Sumby, 1952; Fritz & Grier, 1955).

1.5 Some Connections with Grammaticalness

In Sec. 3 of Chapter 11 we mentioned the difficult problem of assigning degrees of grammaticalness to strings in a way that would reflect the

manner and extent of their deviation from well-formedness in a given language. Some of the concepts introduced in the present chapter suggest a possible approach to this problem.[3]

Suppose we have a grammar G that generates a fairly narrow (though, of course, infinite) set $L(G)$ of well-formed sentences. How could we assign to each string not generated by the grammar a measure of its deviation in at least one of the many dimensions in which deviation can occur? We might proceed in the following way: select some unit—for concreteness, let us choose word units and, for convenience, let us not bother to distinguish in general between different inflectional forms (e.g., between *find, found, finds*). Next, set up a hierarchy $\mathscr{C}$ of classes of these units, where $\mathscr{C} = \mathscr{C}_1, \ldots, \mathscr{C}_N$, and for each $i \leqslant N$

$$\mathscr{C}_i = \{C_1^i, \ldots, C_{a_i}^i\}, \text{ where: } a_1 > a_2 > \ldots > a_N = 1,$$
$$C_j^i \text{ is nonnull,}$$
$$\text{for each word } w, \text{ there is a } j \text{ such that } w \in C_j^i,$$
$$\text{and } C_j^i \subseteq C_k^i \text{ if and only if } j = k. \quad (22)$$

$\mathscr{C}_1$ is the most highly differentiated class of categories; $\mathscr{C}_N$ contains but a single category. Other conditions might be imposed (e.g., that $\mathscr{C}_i$ be a refinement of $\mathscr{C}_{i+j}$), but Condition 22 suffices for the present discussion.

$\mathscr{C}_i$ is called the *categorization of order i*; its members are called *categories of order i.* A sequence $C_{b_1}^i, \ldots, C_{b_q}^i$ of categories of order i is called a *sentence-form* of order i; it is said to generate the string of words $w_1 \ldots w_q$ if, for each $j \leqslant q$, $w_j \in C_b^i$. Thus the set of all word strings generated by a sentence-form is the complex (set) product of the sequence of categories.

We have described $\mathscr{C}$ and G independently; let us now relate them. We say that a set Σ of sentence-forms of order i *covers* G if each string of $L(G)$ is generated by some member of Σ. We say that a sentence-form is *grammatical* with respect to G if one of the strings that the sentence-form generates is in $L(G)$—*fully grammatical*, with respect to G, if each of the strings that it generates is in $L(G)$. We say that $\mathscr{C}$ is *compatible* with G if for each sentence w of $L(G)$ there is a sentence-form of order one that generates w and that is fully grammatical with respect to G. Thus, if $\mathscr{C}$ is compatible with G, there is a set of fully grammatical sentence-forms of order one that covers G. We might also require, for compatibility, that $\mathscr{C}_1$ be the smallest set of word classes to meet this condition. Note in this case that the categories of $\mathscr{C}_1$ need not be pairwise disjoint. For example,

[3] The idea of using information measures to determine an optimal set of syntactic categories, as outlined here, was suggested by Peter Elias. This approach is developed in more detail, with some supporting empirical evidence, in Chomsky (1955, Chapter 4).

know will be in C_i^1 and *no* in C_j^1, where $i \neq j$, although they are phonetically the same. If two words are mutually substitutable throughout $L(G)$, they will be in the same category C_j^1, if it is compatible with G, but the converse is not necessarily true.

We say that a string w is *i-grammatical* (has degree of grammaticalness i) with respect to G, $\mathscr{C}$ if i is the least number such that w is generated by a grammatical sentence-form of order i. Thus the strings of the highest degree of grammaticalness are those of order 1, the order with the largest number of categories. All strings are grammatical of order N or less, since $\mathscr{C}_N$ contains only one category.

These ideas can be clarified by an example. Suppose that G is a grammar of English and that $\mathscr{C}$ is a system of categories compatible with it and having a structure something like this:

$$\begin{aligned}
\mathscr{C}_1\colon\ N_{\text{hum}} &= \{\text{boy, man, } \ldots\} \\
N_{\text{ab}} &= \{\text{virtue, sincerity, } \ldots\} \\
N_{\text{comp}} &= \{\text{idea, belief, } \ldots\} \\
N_{\text{mass}} &= \{\text{bread, beef, } \ldots\} \\
N_{\text{comm}} &= \{\text{book, chair, } \ldots\} \\
V_1 &= \{\text{admire, dislike, } \ldots\} \\
V_2 &= \{\text{annoy, frighten, } \ldots\} \\
V_3 &= \{\text{hit, find, } \ldots\} \\
V_4 &= \{\text{sleep, reminisce, } \ldots\} \\
&\text{etc.} \\
\mathscr{C}_2\colon\ \text{Noun} &= N_{\text{hum}} \cup N_{\text{ab}} \cup \ldots \\
\text{Verb} &= V_1 \cup V_2 \cup \ldots \\
&\text{etc.} \\
\mathscr{C}_3\colon\ &\text{Word.}
\end{aligned} \tag{23}$$

This extremely primitive hierarchy $\mathscr{C}$ of categories would enable us to express some of the grammatical diversity of possible strings of words. Let us assume that G would generate *the boy cut the beef, the boy reminisced, sincerity frightens me, the boy admires sincerity, the idea that sincerity might frighten you astonishes me, the boy found a piece of bread, the boy found the chair, the boy who annoyed me slept here*, etc. It would not, however, generate such strings as *the beef cut sincerity, sincerity reminisced, the boy frightens sincerity, sincerity admires the boy, the sincerity that the idea might frighten you astonishes me, the boy found a piece of book, the boy annoyed the chair, the chair who annoyed me found here*, etc. Strings of

the first type would be one-grammatical (as are all strings generated by G); strings of the second type would be two-grammatical; all strings would be three-grammatical, with respect to this primitive categorization.

Many of the two-grammatical strings might find a natural use in actual communication, of course. Some of them, in fact, (e.g., *misery loves company*, etc.) might be more common than many one-grammatical strings (an infinite number of which have zero probability and consist of parts which have zero probability, effectively).

A speaker of English can impose an interpretation on many of these strings by considering their analogies and resemblances to those generated by the grammar he has mastered, much as he can impose an interpretation on an abstract drawing. One-grammatical strings, in general, like representational drawings, need have no interpretation *imposed* on them to be understood. With a hierarchy such as $\mathscr{C}$ we could account for the fact that speakers of English know for example, that *colorless green ideas sleep furiously* is surely to be distinguished, with respect to well-formedness, from *revolutionary new ideas appear infrequently* on the one hand and from *furiously sleep ideas green colorless* or *harmless seem dogs young friendly* (which has the same pattern of grammatical affixes) on the other; and so on, in an indefinite number of similar cases.

Such considerations show that a generative grammar could more completely fulfil its function as an explanatory theory if we had some way to project, from the grammar, a certain compatible hierarchy $\mathscr{C}$ in terms of which degree of grammaticalness could be defined. Let us consider now how this might be done.

In order to simplify exposition, we first restrict the problem in two ways. We shall consider only sentences of some fixed length, say length λ. Second, let us consider the problem of determining the system of categories $\mathscr{C}_i = \{C_1^i, \ldots C_{a_i}^i\}$, where a_i is fixed. The best choice of a_i categories is the one that in the appropriate sense maximizes substitutability relations among the categorized elements. The question, then, is how we can select the fixed number of categories which best mirror substitutability relations. Note that we are interested in substitutability not with respect to $L(G)$ but to contexts stated in terms of the categories of $\mathscr{C}_i$ itself. To take an example, *boy* and *sincerity* are much more freely substitutable in contexts defined by the categories of $\mathscr{C}_2$ of (23) than in actual contexts of $L(G)$; thus we may find both words in the context Noun Verb Determiner——, but not in the context *you frightened the* ——. Some words may not be substitutable at all in $L(G)$, although they are mutually substitutable in terms of higher order categories. This fact suggests that systematic procedures of substitution applied to successive words in some sequence of grammatical sentences will probably always fail—as, indeed, they always

have so far—since the maximization of substitutability, in the sense we intend here, is a property of the whole system of categories.

A better way to approach the problem is this: suppose that σ_1 is a sequence $s_1, \ldots, s_m$ of all sentences of length λ in $L(G)$ and that $\mathscr{C}_i$ is a proposed set of a_i categories. Let σ_2 be a sequence of sentence-forms $\Sigma_1, \ldots, \Sigma_m$, where, for each $j \leqslant m$, Σ_j generates s_j and Σ_j consists of categories of $\mathscr{C}_i$. There will, of course, be many repetitions in σ_2, in general. Let σ_3 be the sequence $t_1, \ldots, t_n$ of all strings generated by the Σ_j's in σ_2, where σ_3 contains no repetitions. For example, if σ_1 contains *the boy slept* and *the period elapsed*, but not *the period slept* or *the boy elapsed*, and if σ_2 is based on a categorization into nouns and verbs [i.e., σ_2 contains (Determiner, Noun, Verb)], then σ_3 would contain all four of those sentences.

It seems reasonable to measure the adequacy of a system of categories by some function of the length of the sequence σ_3. The number of generated sentences in σ_3 indicates the extent to which the categorization reflects substitutability relations not only with respect to the given set of sentences but also with respect to contexts defined in terms of the categories themselves. Thus particular nouns may not be substitutable with respect to the same verbs, but they do each occur in a given position relative to some verb so that they are substitutable with respect to the category Verb. The same is true of particular verbs, adjectives, etc. This approach permits us to set up all the categories simultaneously.

To evaluate a system $\mathscr{C}_i$ of a_i categories, given a sequence σ_1 of actual sentences of length λ, we shall try to discover a sequence σ_2 that covers σ_1, in the sense we have defined (more precisely, whose terms constitute a set that covers σ_1), and that is *minimal* in the sense that it generates the shortest sequence σ_3. In case the categories of $\mathscr{C}_i$ are pairwise disjoint, this procedure is perfectly simple; we merely replace each word in the strings of σ_1 by the category of $\mathscr{C}_i$ to which it belongs, thus forming σ_2. But, if the categories of $\mathscr{C}_i$ overlap, there may be many covering sequences σ_2; we must find the minimal one in order to evaluate $\mathscr{C}_i$.

Categories overlap in the case of grammatical homonyms, as we have observed. Note that if a word is put into more than one category when we form $\mathscr{C}_i$ the value of this categorization will always suffer a loss in one respect. Each time a category appears in a sentence-form of σ_2 a set of sentences of σ_3 is generated for each word in that category. Hence the more words in a category, the more sentences generated and the less satisfactory the categorization. However, if the word assigned to two categories is a bona fide homonym, there may be a compensating saving. Suppose, for example, that the phoneme sequence /nō/ (*know*, *no*) is put only into the category of verbs. Then all verbs will be generated in σ_3 in

the position *there are —— books on the table.* Similarly, if it is put only into the category of determiners, all will be generated in such contexts as *I —— that he has been here.* If /nō/ is assigned to both categories, a given occurrence of /nō/ in σ_1 can be assigned to either verb or determiner. Since verbs will appear anyway in the context *I —— that he has been here* and determiners in *there are —— books on the table,* no new sentence forms are produced by assignment of /nō/ to verb in the first case and to determiner in the second. There is thus a considerable saving in the sequence σ_3 of generated strings.

These observations suggest a way to decide when an element should in fact be considered a set of grammatical homonyms. We make this decision when the loss incurred automatically in assigning it to several categories is more than compensated for by the gain that can be achieved through the extra freedom in choosing the complete covering sequence σ_2; there is always a numerical answer to this question. It must be shown, of course, that in terms of presystematic criteria, the solution of the homonym problem given by this approach is the correct one. Certain preliminary investigations of this have been hopeful (cf. Chomsky, 1955), but the task of evaluating and improving this or any other conception of syntactic category is an immense one. Furthermore, several important distinctions have been blurred in this brief discussion.

Let us now return to our two assumptions: (*a*) that the length λ of sentences is fixed and (*b*) that the number a_i of categories is fixed. The first is easily dispensable. Given G, we can evaluate a set $\mathscr{C}_i = \{C_1^i, \ldots, C_{a_i}^i\}$, where a_i is a fixed integer, in the following way. Select a new "word" # to indicate sentence boundary, $\# \notin C_j^i$ for any j. Define a *discourse* as a sequence of words $\#, w_1^1, \ldots, w_{\alpha_1}^1, \#, w_1^2, \ldots, w_{\alpha_2}^2, \#, \ldots, \#, w_1^k, \ldots, w_{\alpha_k}^k$, where for each j, $w_1^j \ldots w_{\alpha_j}^j$ is a sentence of the language generated by G. This is a discourse of length $\alpha_1 + \ldots + \alpha_k + k$. An *initial discourse* is an initial subsequence of a discourse. A *discourse form* is a sequence of categories $C_{\beta_1}^i, \ldots, C_{\beta_q}^i$ of categories of $\mathscr{C}_i$ or $\{\#\}$ such that there is a discourse $w_1, \ldots, w_q$, where, for each j, $w_j \in C_{\beta_j}^i$, and an *initial discourse form* is an initial subsequence of a discourse form. Let Σ_λ be a set of initial discourse forms, each of length λ, which covers the set of initial discourses of length λ and is minimal from the point of view of generation, and let $N(\lambda)$ be the number of distinct strings generated by the members of Σ_λ. Then the natural way to define the value of the categorization $\mathscr{C}_i$ is, by analogy with the definition of channel capacity in Eq. 1, p. 431, as

$$\text{Val}(\mathscr{C}_i) = \lim_{\lambda \to \infty} \frac{\log N(\lambda)}{\lambda}. \tag{24}$$

We choose as the best categorization into a_i categories that analysis $\mathscr{C}_i$ for which Val $(\mathscr{C}_i)$ is minimal. In other words, we select the categorization into a_i categories that minimizes the information per word, that is, maximizes the redundancy, in the generated "language" of grammatical discourses (assuming independence of successive sentences). Thus we shall try to select a categorization that maximizes the contribution of the category analysis to the total set of constraints under which the source operates in producing discourses. In practice, this computation can be much simplified by assuming that successive choices of Σ_λ, for increasing λ, are not independent.

We have now proposed a definition of optimal categorization into n categories, for each n, which is independent of arbitrary decisions about sentence length. We must finally consider the assumption that we are given the integers $a_1, \ldots, a_N$ which determine the number of categories in Condition 22. Suppose, in fact, that we determine for each n the optimal categorization K_n into n categories, in the way previously sketched. To select from the set $\{K_n\}$ the hierarchy $\mathscr{C}$, we must determine for which integers a_i we will actually set up the optimal categorization K_{a_i} as an order $\mathscr{C}_i$ of $\mathscr{C}$. We would like to select a_i in such a way that K_{a_i} will be clearly preferred to K_{a_i-1} but will not be much worse than K_{a_i+1}; that is to say, we would like to select K_{a_i} in such a way that there will be a considerable loss in forming a system of categories with fewer than a_i categories but not much of a gain in adding a further category.

We might, for example, take $\mathscr{C}_i = K_{a_i}$ as an order of $\mathscr{C}$ just in case the function $f(n) = n\text{Val}\,(K_n)$ has a relative minimum at $n = a_i$. (We might also be interested in the absolute minimum of f, defined in this or some more appropriate way—we might take this as defining an absolute order of grammaticalness and an overriding bifurcation of strings into grammatical and ungrammatical, with the grammatical including as a proper subclass those generated by the grammar.)

In the way just sketched we might prescribe a general procedure Ψ such that, given a grammar G, $\Psi(G)$ is a hierarchy $\mathscr{C}$ of categories compatible with G, by which degree of grammaticalness is defined for each string in the terminal vocabulary of G. It would then be correct to say that a grammar not only generates sentences with structural descriptions but also assigns to each string, whether generated or not, a degree of grammaticalness that measures its deviation from the set of perfectly well-formed sentences as well as a partial structural description that indicates how this string deviates from well-formedness.

It is hardly necessary to emphasize that this proposal is, in its details, highly tentative. Undoubtedly there are many other ways to approach this complex question.

1.6 Minimum-Redundancy Codes

Before a message can be transmitted, it must be coded in a form appropriate to the medium through which it will pass. This coding can be accomplished in many ways; the procedure becomes of some theoretical interest, however, when we ask about its efficiency. For a given alphabet, what codes will, on the average, give the shortest encoded messages? Such codes are called *minimum-redundancy codes*. Natural languages are generally quite redundant; how to encode them to eliminate that redundancy poses a challenging problem.

The question of coding efficiency becomes especially interesting when we recognize that every channel is noisy, so that an efficient code must not only be short but at the same time must enable us to keep erroneous transmissions below some specified probability. The solutions that have been found for this problem constitute the real core of information theory as it is applied to many practical problems in communication engineering. Inasmuch as psychologists and linguists have not yet exploited these fundamental results for noisy channels, we shall limit our attention here to the simpler problem of finding minimum-redundancy codes for noiseless channels.

The problem of optimal coding can be posed as follows: we know from Sec. 1.3 that an alphabet is used most efficiently when each character occurs independently and equiprobably, that is, when all strings of equal length are equiprobable. So we must find a function θ that maps our natural messages into coded forms in which all sequences of the same length are equiprobable. For the sake of simplicity, let us assume that the messages can be divided into independent units that can be separately encoded. In order to be definite, let us imagine that we are dealing with printed English and that we are willing to assume that successive words are independent. Each time a space occurs in the text the text accumulated since the preceding space is encoded as a unit. For each word, therefore, we shall want to assign a sequence of code symbols in such a way that, on the average, all the code symbols will be used independently and equally often and in such a way that we shall be able to segment the coded messages to recover the original word units when the time comes to decode it.

First, we observe that in any minimum-redundancy code the length of a given coded word can never be less than the length of a more probable coded word. If the more probable word were longer, a saving in the average length could be achieved by simply reversing the codes assigned to the two words. We begin, therefore, by ranking the words in order of decreasing probability of occurrence. Let p_r represent the probability of

the word ranked r, and let c_r represent the length of its encoded representation; that is to say, we rank the words

$$p_1 \geqslant p_2 \geqslant \ldots \geqslant p_{N-1} \geqslant p_N,$$

where N is the number of different words in the vocabulary. For a minimum redundancy code we must then have

$$c_1 \leqslant c_2 \leqslant \ldots \leqslant c_{N-1} \leqslant c_N.$$

Note, moreover, that the mean length C of an encoded word will be

$$C = \sum_{r=1}^{N} p_r c_r. \tag{25}$$

Obviously, the mean length would be a minimum if we could use only one-letter words, but this would entail too large a number D of different code characters. Ordinarily, our choice of D is limited by the nature of the channel. Of course, it is not length per se that we want to minimize but length per unit of information transmitted. The problem is to minimize C/H, the length per bit (or to maximize H/C, the amount of information per unit length), subject to the subsidiary conditions that $\Sigma p_r = 1$ and that the coded message be uniquely decodable.

By virtue of Assumption 2 in Sec. 1.3 it would seem that H/C, the information per letter in the encoded words, cannot be greater than $\log D$, the capacity of the coding alphabet. From that fact we might try to move directly to a lower bound,

$$C \geqslant \frac{H}{\log D}. \tag{26}$$

Although this inequality is correct, it cannot be derived as a simple consequence of Assumption 2. Consider the following counter-example (Feinstein, 1958): we have a vocabulary of three words with probabilities $p_1 = p_2 = 2p_3 = 0.4$ and we code them into the binary alphabet $\{0, 1\}$ so that $\theta(1) = 0$, $\theta(2) = 1$, and $\theta(3) = 01$. Now we can easily compute that $C = 1.2$, $H = 1.52$, and $\log_2 D = 1$, so that the average length is less than the bound stated in Eq. 26. The trouble, of course, is that θ does not yield a true code, in the sense defined in Chapter 11; the coded messages are not uniquely decodable. If, however, we add to Assumption 2 the further condition of unique decodability, the lower bound stated in Eq. 26 can be established. The further condition is most easily phrased in terms of a left tree code, in which no coded word is an initial segment of any other coded word. By using Eq. 21 we can write

$$H = -\sum_{i=1}^{N} p_i \log p_i \leqslant -\sum_{i=1}^{N} p_i \log \frac{D^{-c_i}}{\sum_{l=1}^{N} D^{-c_l}} = \log \sum_{l=1}^{N} D^{-c_l} + \sum_{i=1}^{N} p_i c_i \log D.$$

For left tree codes we know, from Eq. 4, in Chapter 11, that $\Sigma D^{-c_i} \leqslant 1$; therefore,

$$\log \sum D^{-c_i} \leqslant \log 1 = 0,$$

so we can write

$$H \leqslant \sum_{i=1}^{N} p_i c_i \log D,$$

from which the desired inequality of Eq. 26 follows by rearranging terms.

If Eq. 26 sets a lower bound on the mean length C, how closely can we approach it? The following theorem, due to Shannon (1948), provides the answer:

Theorem 4. *Given a vocabulary V of N words with information H and a coding alphabet A of D code symbols, it is possible to code the words by finite strings of code symbols from A in such a way that C, the average number of code symbols per word, satisfies the inequality*

$$\frac{H}{\log D} \leqslant C < \frac{H}{\log D} + 1. \tag{27}$$

The proof has been published in numerous places; see, for example, Feinstein (1958, Chapter 2) or Fano (1961, Chapter 3).

Instead of proving here that such minimum-redundancy codes exist, we shall consider ways of constructing them. Both Shannon (1948) and Fano (1949) proposed methods of constructing codes that approach minimum redundancy asymptotically as the length of the coding unit is enlarged to include progressively longer sequences of words. In 1952, however, Huffman discovered a method of systematically constructing minimum-redundancy codes for finite vocabularies without resorting to any limiting operations.

Huffman assumes that the vocabulary to be encoded is finite, that the probability of each word is known in advance, that a left tree code can be used, and that all code symbols will be of unit length. Within these limits, let us now consider the special conditions that a minimum-redundancy code must satisfy:

1. No two words can be represented by identical strings of code symbols.
2. It must be possible for a receiver to segment coded messages into the coded words that comprise them. (This restriction is discussed in Chapter 11, Sec. 2.) The printer's use of a special symbol (space) to mark word boundaries in a natural code is in general too inefficient for minimum redundancy codes. Proper segmentation in the sense of boundary markers is ensured, however, by the assumption that it must be a left tree code.
3. If the words are ranked in order of decreasing probability p_r, then the length of the rth word, c_r, must satisfy the inequalities

$$c_1 \leqslant c_2 \leqslant \ldots \leqslant c_{N-1} = c_N.$$

Because all the code symbols are equally long, c_r can be interpreted simply as the number of symbols used to code the rth word. In a minimum redundancy tree code $c_{N-1} = c_N$ because the first c_{N-1} symbols used to code the Nth word cannot be the coded form of any other word; that is to say, the coded forms of words $N - 1$ and N must differ in their first c_{N-1} symbols, and, if they do, no additional symbols are needed to encode word N.

4. At least two (and not more than D) words of code length c_N have codes that are identical except for their final digits. Imagine a minimum redundancy tree code in which this was not true; then the final code symbols could be deleted, thus shortening the average length of a coded word and so leading to a contradiction.

5. Each possible string of $c_N - 1$ code symbols must be used either to represent a word or some initial segment of the sequence must represent a word. If such a string of symbols existed and was not used, the average length of the coded words could be reduced by using it in place of some longer string.

These restrictions are sufficient to determine the following procedure, which we shall outline for a binary coding alphabet, $D = 2$. List the words from most probable to least probable. By (3), $c_{N-1} = c_N$, and, by (4), there are exactly two words of code length c_N that must be identical except for their final symbols. So we can assign 0 as the final digit of the $(N - 1)$th word and 1 as the final digit of the Nth word. Once this has been done, the $(N - 1)$th and Nth words taken together are equivalent to a single composite message; its code will be the common (but still unknown) initial segment of length $c_N - 1$ and its probability will be the sum of the probabilities of the two words comprising it. By combining these two words, we create a new vocabulary with only $N - 1$ words in it. Suppose we now reorder the words as before and repeat the whole procedure. We can continue to do so until the reduced vocabulary contains only two words, at which point we assign 0 to one and 1 to the other and the code is completed.

An illustration of this procedure, using a binary code, is shown in Table 2. A vocabulary of nine words is given in order of decreasing probability. The first step is to assign 0 to word h and 1 to word i (or conversely) as their final code symbols, then to combine h and i into a single item in a new derived distribution. The procedure is then repeated for the two least probable items in this new distribution, etc., until all the code symbols have been assigned. The result is to produce a coding tree; it can be seen with difficulty in Table 2, in which its trunk is on the right and its branches extend to the left, or more easily in Fig. 2, in which it has been redrawn in the standard way.

Table 2 Huffman's Method of Constructing a Minimum-Redundancy Code

Word	Coded Form	Original Distribution	First Derived Distribution	Second	Third	Fourth	Fifth	Sixth	Seventh	Final
a	01	0.27	0.27	0.27	0.27	0.27	0.30	0.43	0.57	1.00
b	10	0.23	0.23	0.23	0.23	0.23	0.27	0.30	0.43	
c	000	0.15	0.15	0.15	0.15	0.20	0.23	0.27		
d	110	0.10	0.10	0.10	0.15	0.15	0.20			
e	0010	0.08	0.08	0.10	0.10	0.15				
f	0011	0.07	0.07	0.08	0.10					
g	1110	0.05	0.05	0.07						
h	11110	0.03	0.05							
i	11111	0.02								

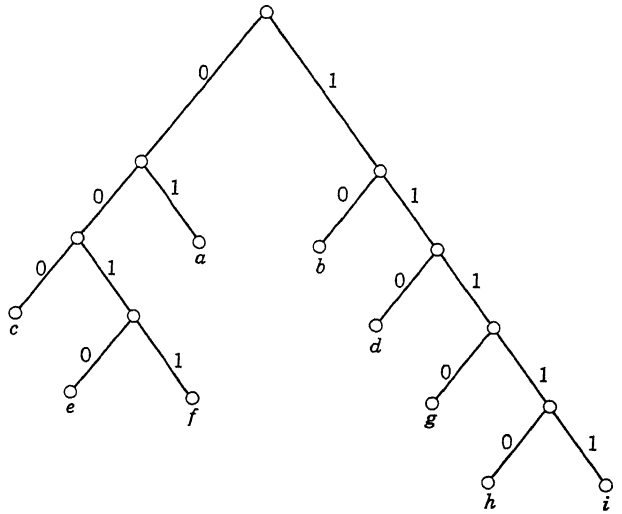


Fig. 2. The coding tree developed for the minimum-redundancy code of Table 2.

In order to evaluate the coding efficiency, we need to know $\log D$, C, and H. The coding alphabet is binary, $D = 2$, and its information capacity is $\log_2 D = 1$ bit per symbol. The average length of an encoded word can be easily computed from Table 2 by Eq. 25; the result is 2.80 binary code symbols per word. The amount of information in the original distribution or word probabilities can be computed by Eq. 13; the result is 2.781 bits per word. In terms of Theorem 4, therefore, we have

$$\frac{2.78}{1} \leqslant 2.80 < \frac{2.78}{1} + 1,$$

which indicates that for this example the average length is already quite close to its lower bound. The redundancy of the coded signal—as defined by Eq. 17—is less than 1%.

It should be obvious that errors in the transmission or reception of minimum-redundancy codes are difficult to detect. Every branch of the coding tree is utilized and errors convert any intended message into a perfectly plausible alternative message. Considerable study has been devoted to the most efficient ways to introduce redundancy into the code deliberately in order to make errors easier to detect and to correct. But these artificially redundant codes are not surveyed here. The point should be noted, however, that the redundancy of natural codes may not be so inefficient as it seems, for it can help us to communicate under less than optimal conditions.

The reason that minimum-redundancy codes are important is an economic one. There is a cost to communication and someone must pay for it. It is often appropriate to use C, the average length of the message, as a measure of the cost, since it takes either more time or more equipment to transmit more symbols. It should be recognized, however, that the economy achieved by minimizing C/H affects the supply price, not the demand price of this commodity (Marschak, 1960). The supply price is the lowest price the supplier is willing to charge; the demand price is the highest price the buyer is willing to pay. The demand price depends on the payoff that the customer expects to obtain by using the information; since that use will ordinarily involve the meaning of the message in an essential way, it takes us beyond the limits we have arbitrarily imposed on this chapter.

1.7 Word Frequencies

It is scarcely surprising to find that the various words of a natural language do not occur equally often in any reasonable sample of ordinary discourse. Some words are far more common than others. Psychologists have recognized the importance of these unequal probabilities for any kind of experimentation that uses words as stimuli. It is standard procedure for psychologists to try to control for the effects of unequal familiarity by selecting the words from some tabulation of relative frequencies of occurrence. For English the Thorndike-Lorge (1944) counts are probably the best known and most widely used. An extensive technical literature deals with the various statistics that have been compiled for the (usually written) languages of the world; we shall make no attempt to review or evaluate it in these pages. Instead, we shall concentrate our attention on certain statistical aspects of the vocabulary that seem theoretically most significant.

There is one particularly striking regularity that has been found in these various statistical explorations. The following is perhaps the simplest way to summarize it (Mandelbrot, 1959): consider a (finite or infinite) population of discrete *items*, each of which carries a *label* chosen from a discrete set. Let $n(f, s)$ be the number of different labels that occur exactly f times in a sample of s items. Then one finds that, for large s,

$$n(f, s) = G(s)f^{-(1+\rho)}, \tag{28}$$

where $\rho > 0$ and $G(s)$ is a constant depending on the size of the sample.

If Eq. 28 is expressed as a probability density, then it is readily seen that the variance of f is finite if and only if $\rho > 2$ and that the mean of f is finite if and only if $\rho > 1$. In the cases of interest in this section it is often

true that $\rho < 1$, so we are faced with a law that is often judged anomalous (or even pathological) to those prejudiced in favor of the finite means and variances of normal distributions. In the derivation of the normal distribution function, however, it is necessary to assume that we are dealing with the sum of a large number of variables, each of which makes a small contribution relative to the total. When equal contributions are not assumed, however, it is still possible to have stable limit distributions, but either the second moment (and all higher moments) will be infinite, or all moments will be infinite (cf. Gnedenko & Kolmogorov, 1954, Chapter 7). Such is the distribution underlying Eq. 28.

Nonnormal limit distributions might be dismissed as mathematical curiosities of little relevance were it not for the fact that they have been observed in a wide variety of situations. As Mandelbrot (1958) has pointed out, these situations seem especially common in the social sciences. For example, if the items are quantities of money and the labels are the names of people who earn each item, then $n(f, s)$ will be the number of people earning exactly f units of money out of a total income equal to s. In this form the law was first stated (with $\rho > 1$) by Pareto (1897). Alternatively, if the items are taxonomic species and the labels are the names of genera to which they belong, then $n(f, s)$ will be the number of genera each with exactly f species. In this form the law was first stated by Willis (1922), then rationalized by Yule (1924), with $\rho < 1$ (and usually close to 0.5).

In the present instance, if the items are the consecutive words in a continuous discourse by a single author and the labels are sequences of letters used to encode words, then $n(f, s)$ will be the number of letter sequences (word types) that occur exactly f times in a text of s consecutive words (word tokens). In this form the law was first stated by Estoup (1916), rediscovered by Condon (1928), and intensively studied by Zipf (1935). Zipf believed that $\rho = 1$, but further analysis has indicated that usually $\rho < 1$. Considerable data indicating the ubiquity of Eq. 28 were provided by Zipf (1949), and empirical distributions of this general type have come to be widely associated with his name.

When working with word frequencies, it is common practice to rank them in order (as we did for the coding problem in the preceding section) from the most frequent to the least frequent. The rank r is then defined as the number of items that occur f times or more:

$$r = \sum_{j=f}^{\infty} n(j, s).$$

If we combine this definition with Eq. 28 and approximate the sum by an integral, then, for large f,

$$r \sim \frac{G(s)}{\rho f^{\rho}},$$

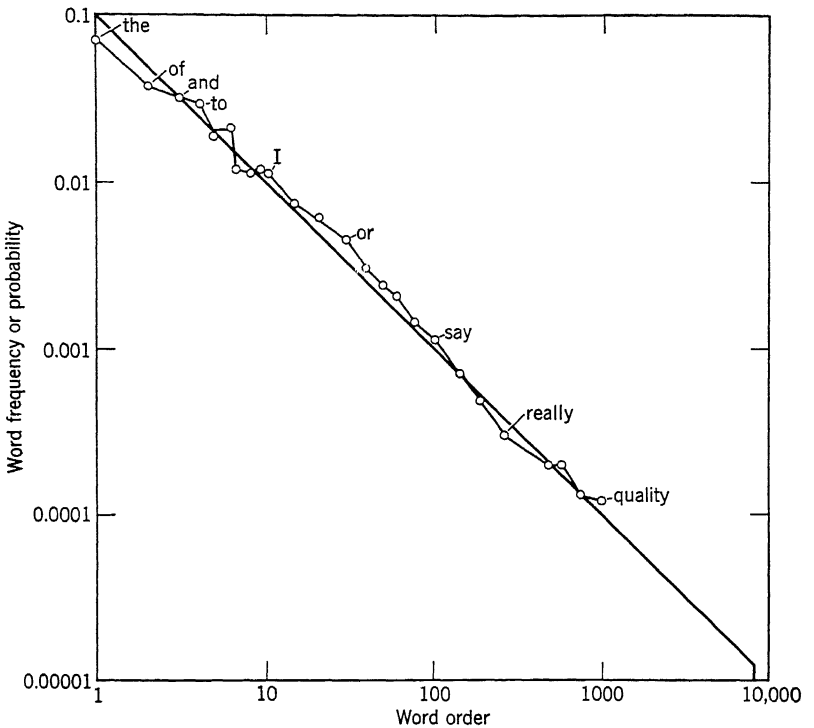


Fig. 3. The rank-frequency relation plotted on log-log coordinates.

which states a reciprocal relation between the ranks r and the frequencies f. We can rewrite this relation as

$$f \sim \left(\frac{G(s)}{\rho r}\right)^{1/\rho} = K' r^{-B}, \tag{29}$$

where $B = 1/\rho$. Therefore

$$\log f \sim K - B \log r,$$

which means that on log-log coordinates the rank-frequency relation should give a straight line with a slope of $-B$. It was with such a graph that the law was discovered, and it is still a popular way to display the results of a word count. An illustration is given in Fig. 3.

The persistent recurrence of stable laws of this nonnormal type has stimulated several attempts at explanation, and there has been considerable discussion of their relative merits. We shall not review that discussion here; the present treatment follows Mandelbrot (1953, 1957) but does little more than introduce the topic in terms of its simplest cases.

Imagine that the coded message is, in fact, a table of random (decimal)

digits. Let the digits 0 and 1 play the role of word-boundary markers; each time 0 or 1 occurs it marks the beginning of a new word. (In this code there are words of zero length; a minor modification can eliminate them if they are considered anomalous.) The probability of getting a particular word of exactly length i is (probability of symbol)^i(probability of boundary marker) = $(0.8)^i(0.2)$, and the number of different words of length i is 8^i.

The critical point to note in this example is that when we order these coded words with respect to increasing length we have simultaneously ordered them with respect to decreasing probability. Thus it is possible to construct Table 3. The one word of zero length has a probability of 0.2

Table 3 The Rank-Frequency Relation for a Random Code

Length i	Probability $p(w_i)$	Number D^i	Ranks ΣD^i	Average Rank $r(w_i)$
0	0.2	1	1	1
1	0.02	8	2–9	5.5
2	0.002	64	10–73	41.5
3	0.0002	512	74–585	329.5
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.

and, since it is the most probable word, it receives rank 1. The eight words one digit long all have a probability of 0.02 and share ranks 2 through 9; we assign them all the average rank 5.5; and so the table continues. When we plot these values on log-log coordinates, we obtain the function shown in Fig. 4. Visual inspection indicates that the slope is slightly steeper than -1, which is also characteristic of many natural-language texts.

It is not difficult to obtain the general equation relating probability to average rank for this simple random case (Miller, 1957). Let $p(\#)$ be the probability of a word-boundary marker, and let $1 - p(\#) = p(L)$ be the probability of a letter. If the alphabet (excluding #) has D letters, then $p(L)/D$ is the probability of any particular letter, and $p(w_i) = p(\#)p(L)^iD^{-i}$ is the probability of any particular word of length i (= 0, 1, . . .). This quantity will prove to be more useful when written

$$\begin{aligned} p(w_i) &= p(\#)e^{-i\log D}e^{i\log p(L)} \\ &= p(\#)(e^{i\log D})^{-\left(1-\frac{\log p(L)}{\log D}\right)} \\ &= p(\#)(e^{i\log D})^{-B}. \end{aligned} \tag{30}$$

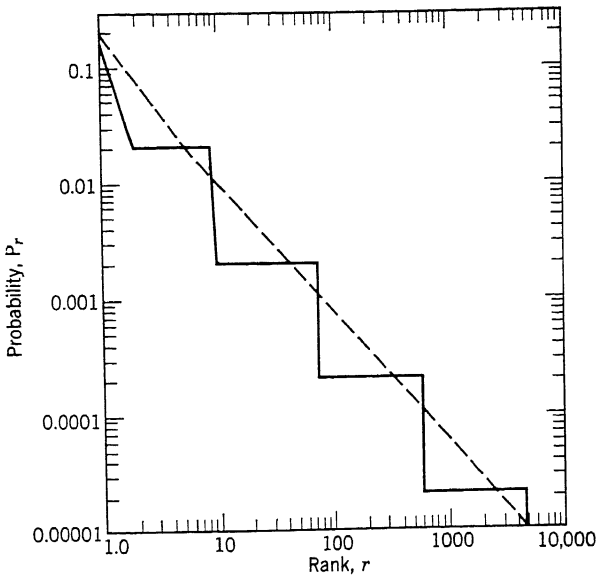


Fig. 4. The rank-frequency relation for strings of random digits occurring between successive occurrences of 0 or 1. The solid line represents the expected function and the dashed line representsthe average ranks.

Since there are D^j different words of exactly length j, there must be $\sum_{j=0}^{j=i} D^j$ of them equal to or shorter than i, so that when we rank them in order of increasing length the D^j words of length j will receive ranks $1 + \sum_{j=0}^{j=i-1} D^j$ to $\sum_{j=0}^{j=i} D^j$. The average rank will be

$$\begin{aligned} r(w_i) &= \frac{1}{2}\left(1 + \sum_{0}^{i-1} D^j + \sum_{0}^{i} D^j\right) \\ &= \frac{1}{2}\left(1 + \frac{D^i - 1}{D - 1} + \frac{D^{i+1} - 1}{D - 1}\right) \\ &= D^i \frac{D + 1}{2(D - 1)} + \frac{D - 3}{2(D - 1)}, \end{aligned}$$

which will prove more useful if we write

$$D^i = e^{i \log D} = \frac{2(D-1)}{D+1}\left[r(w_i) - \frac{D-3}{2(D-1)}\right] = \frac{2(D-1)}{D+1}[r(w_i) - c], \tag{31}$$

for now Eqs. 30 and 31 combine to give

$$p(w_i) = p(\#)\left\{\frac{2(D-1)}{D+1}\,[r(w_i) - c]\right\}^{-B} = K'[r(w_i) - c]^{-B}, \quad (32)$$

which can be recognized as a variant form of Eq. 29, where

$$B = 1 - \frac{\log p(L)}{\log D}, \qquad c = \frac{D-3}{2(D-1)}, \quad \text{and} \quad K' = p(\#)\left[\frac{2(D-1)}{D+1}\right]^{-B}.$$

Thus a table of random numbers can be seen to follow the general type of law that has been found for word frequencies. If we take $D = 26$ and $p(\#) = 0.18$ to represent written English, then

$$B = 1 - \frac{\log 0.82}{\log 26} = 1.06,$$

$$c = \frac{26-3}{50} = 0.46,$$

and

$$K' = 0.18\left(\frac{50}{27}\right)^{-1.06} = 0.09,$$

so we have

$$p(w_i) = 0.09\,[r(w_i) - 0.46]^{-1.06}.$$

Since $c = 0.46$ will quickly become negligible as $r(w_i)$ increases, we can write

$$p(w_i) \sim 0.09\,r(w_i)^{-1.06},$$

which is, in fact, close to the function that has been observed to hold for many normal English texts (Zipf, for example, liked to put $K' = 0.1$ and $B = 1$).

The hypothesis that word boundaries occur more or less at random in English text, therefore, has some reasonable consequences. It helps us to understand why the probability of a word decreases so rapidly as a function of its length—which is certainly true, on the average, for English. The critical step in the derivation of Eq. 32, however, occurs when we note that for the random message the rank with respect to increasing length and the rank with respect to decreasing probability are the same. In English, of course, this precise equivalence of rankings does not hold—otherwise we would never let our most frequent word *the* require three letters—but it holds approximately. Miller and Newman (1958) have verified the prediction that the *average* frequency of words of length i is a reciprocal function of their *average* rank with respect to increasing length, where the slope constant for the length-frequency relation on log-log coordinates is close to but perhaps somewhat smaller than B.

In Sec. 1.6 we noted that for a minimum-redundancy code the length of any given word can never be less than the length of a more probable word. Suppose, therefore, that we consider the rank-frequency relation for optimal codes, that is, for codes in which the lower bound on the average length C is actually realized, so that $C = H/\log D$. This optimal condition will hold when the length i of any given word is directly proportional to the amount of information associated with it:

$$i = \rho \frac{-\log p(w_i)}{\log D},$$

where ρ depends on the choice of scale units. This equation can be rewritten as

$$p(w_i) = (e^{i \log D})^{-1/\rho},$$

which is Eq. 30 again, with $B = 1/\rho$. From here on the argument can proceed exactly as before. We see, therefore, that the rank-frequency relation holds quite generally for minimum-redundancy codes because such codes (like tables of random numbers) use all equally long sequences of symbols equally probably. The fact that both minimum-redundancy codes and natural languages (which are certainly far from minimum-redundancy) share the rank-frequency relation in Eq. 29 is interesting, of course, but it provides no basis whatsoever for any speculation that there is something optimal about the coding used in natural languages.

The choice of the digits 0 and 1 as boundary markers to form words in a table of random numbers was completely arbitrary; any other digits would have served equally well. If we generalize this observation to English texts, it implies that we might choose some character other than the space as a boundary marker. Miller and Newman (1958) have studied the rank-frequency relation for a (relatively small) sample of pseudo-words formed by using the letter *E* as the word boundary (and treating the space as just another letter). The null word *EE* was most frequent, followed closely by *ERE*, *E#E*, and so on. As predicted, a function of the general type of Eq. 29 was also obtained for these pseudo-words (but with a slope constant B slightly less than unity, perhaps attributable to inadequate sampling).

There is an enormous psychological difference between the familiar words formed by segmenting on spaces and the apparently haphazard strings that result when we segment on *E*. Segmenting on spaces respects the highly overlearned strings—Miller (1956) has referred to them as chunks of information in order to distinguish sharply from the bits of information defined in Sec. 1.3—that normally function as unitary, psychological elements of language. It seems almost certain, therefore,

that an evolutionary process of selection must have been working in favor of short words—some psychological process that would not operate on the strings of characters between successive *Es*. Thus we find many more very long, very improbable pseudo-words.

In one form or another the hypothesis that we favor short words has been advanced by several students of language statistics. Zipf (1935) has referred to it as the *law of abbreviation*: whenever a long word or phrase suddenly becomes common, we tend to shorten it. Mandelbrot (1961) has proposed that historical changes in word lengths might be described as a kind of random walk. He reasons that the probability of lengthening a word and the probability of shortening it should be in equilibrium, so that a steady state distribution of word lengths could be maintained. If the probability of abbreviation were much greater than the probability of expansion, the vocabulary would eventually collapse into a single word of minimum length. If expansion were more likely than abbreviation, on the other hand, the language would evolve toward a distribution with $B < 1$, and, presumably, some upper bound would have to be imposed on word lengths in order for the series $p(w_i)$ to converge, so that $\Sigma p(w_i) = 1$. It should be noted, however, that the existence of a relation in the form of Eq. 29 does not depend in any essential way on some prior psychological law of abbreviation. The central import of Mandelbrot's earlier argument is that Eq. 29 can result from purely random proccesses. Indeed, if there is some law of abbreviation at work, it should manifest itself as a deviation from Eq. 29—presumably in a shortage of very long, very improbable words, a shortage that would not become apparent until extremely large samples of text had been tabulated.

The occurrence of the rank-frequency relation of Eq. 29 does not constitute evidence of some powerful and universal psychological force that shapes all human communication in a single mold. In particular, its occurrence does not constitute evidence that the signal analyzed must have come from some intelligent or purposeful source. The rank-frequency relation, Eq. 29 has something of the status of a null hypothesis, and, like many null hypotheses, it is often more interesting to reject than to accept.

These brief paragraphs should serve to introduce some of the theoretical problems in the statistical analysis of language. There is much more that might be said about the analysis of style, cryptography, estimations of vocabulary size, spelling systems, content analysis, etc., but to survey all that would lead us even further away from matters of central concern in Chapters 11, 12, and 13.

If one were to hazard a general criticism of the models that have been constructed to account for word frequencies, it would be that they are still far too simple. Unlike the Markovian models that envision D^k parameters,

explanations for the rank frequency relation use only two or three parameters. The most they can hope to accomplish, therefore, is to provide a null hypothesis and to indicate in a qualitative way (perhaps) the kind of systems we are dealing with. They can tell us, for example, that any grammatical rule regulating word lengths must be regarded with considerable suspicion—in an English grammar, at least.

The complexity of the underlying linguistic process cannot be suppressed very far, however, and examples of nonrandom aspects are in good supply. For example, if we partition a random population on the basis of some independent criterion, the same probability distribution should apply to the partitions as to the parent population. If, for example, we partitioned according to whether the words were an odd or an even number of running words away from the beginning of the text or according to whether their initial letters were in the first or the last half of the alphabet, etc., we would expect the same rank-frequency relation to apply to the partitions as to the original population. There are, however, several ways to partition the parent population that look as though they ought to be independent but turn out in fact not to be. Thus, for example, Yule (1944) established that the same distribution does not apply when different categories (nouns, verbs, and adjectives) are taken separately; Miller, Newman, and Friedman (1958) showed a drastic difference between the distributions of content words (nouns, verbs, adjectives, adverbs) and of function words (everything else), and Miller (1951, p. 93) demonstrated that the distribution can be quite different if we consider only the words that occur immediately following a given word, such as *the* or *of*. There is nothing in our present parsimonious theories of the rank-frequency relation that could help us to explain these apparent deviations from randomness.

In an effort to achieve a more appropriate level of complexity in our descriptions of the user, therefore, we turn next to models that take account of the underlying structure of natural languages—models that, for lack of a better name, we shall refer to here as algebraic.

2. ALGEBRAIC MODELS

If the study of actual linguistic behavior is to proceed very far, it must clearly pay more than passing notice to the competence and knowledge of the performing organism. We have suggested that a generative grammar can give a useful and informative characterization of the competence of the speaker-hearer, one that captures many significant and deep-seated aspects of his knowledge of his own language. The question is, therefore, how does he put his knowledge to use in producing a desired sentence or

in perceiving and interpreting the structure of presented utterances? How can we construct a model for the language user that incorporates a generative grammar as a fundamental component? This topic has received almost no study, so we can do little more than introduce a few speculations.

As we observed in the introduction to this chapter, models of linguistic performance can generally be interpreted interchangeably as depicting the behavior of either a speaker or a hearer. For concreteness, in the present sections we shall concentrate on the listener's task and frame our discussion largely in perceptual terms. This decision is, however, a matter of convenience, not of principle.

Unfortunately, the bulk of the experimental research on speech perception has involved the recognition of individual words spoken in isolation as part of a list (cf. Fletcher, 1953) and so is of little value to us in understanding the effects of grammatical structure on speech perception. That such effects exist is clear from the fact that the same words are easier to hear in sentences than in isolation (Miller, Heise, & Lichten, 1951; Miller, 1962a). How these effects are caused, however, is not at all clear.

Let us take as our starting point the sentence-recognizing device introduced briefly in Chapter 11, Sec. 6.4. Instead of a relatively passive process of acoustic analysis followed by identification and symbolic representation, we imagined (following Halle & Stevens, 1959, 1962) an active device that recognizes its input by discovering what must be done in order to generate a signal (in some possibly derived form) to match it. At the heart of this active device, of course, is a component M that contains rules for generating a matching signal. Associated with M would be components to analyze and (temporarily) to store the input, components that reflect various semantic and situational constraints suggested by the context of the sentence, a heuristic component that could make a good first guess, a component to make the comparison of the input and the internally generated signals, and perhaps others. On the basis of an initial guess, the device generates an internal signal according to the rules stored in M and tests its guess against the input signal. If the match is unsatisfactory, the discrepancy is used to make a better guess. In this manner the device proceeds to modify its own internal signal until the match is judged satisfactory or the input is dismissed as unintelligible. The program for generating the matching signal can be taken as the symbolic representation of the input.

If it is granted that such a sentence-recognizer can provide a plausible model for human speech perception, we can take it as our starting point and can proceed to try to specify it more precisely. In particular, the two parts of it that seem to perform the most important functions are the contextual component, which helps to generate a first guess, and the

grammatical component M, which imposes the rules for generating the internal signal. We should begin by studying those two components. Even if it were feasible, a study of the ways contextual information can be stored and brought to bear would lead us far beyond the limits we have placed on this discussion. With respect to M, however, the task seems easier. The way the rules for synthesizing sentences might operate is, of course, very much in our present line of sight.

We are concerned with a finite device M in which are stored the rules of a generative grammar G. This device takes as its input a string x of symbols and attempts to understand it; that is to say, M tries to assign to x a certain structural description $F(x)$—or a set $\{F_1(x), \ldots, F_m(x)\}$ of syntactic descriptions in the case of a sentence x that is structurally ambiguous in m different ways. We shall not try to consider all of those real but obscure aspects of understanding that go beyond the assignment of syntactic structural descriptions to sentences, nor shall we consider the situational or contextual features that may determine which of a set of alternative structural descriptions is actually selected in a particular case. There is no point of principle underlying this limitation to syntax rather than to semantics and to single sentences rather than their linguistic and extra-linguistic contexts—it is simply an unfortunate consequence of limitations in our current knowledge and understanding. At present there is little that can be said, with much precision, about those further questions. [See Ziff (1960) and Katz & Fodor (1962) for discussion of the problems involved in the development of an adequate semantic theory and some of the ways in which they can be investigated].

The device M must contain, in addition to the rules of G, a certain amount of computing space, which may be utilized in various different ways, and it must be equipped to perform logical operations of various sorts. We require, in particular, that M assign a structural description $F_i(x)$ to x only if the generative grammar G stored in the memory of M assigns $F_i(x)$ to x as a possible structural description. We say that the device M *(partially) understands the sentence x in the manner of G* if the set $\{F_1(x), \ldots, F_m(x)\}$ of structural descriptions provided by M with input x is (included in) the set assigned to x by the generative grammar G. In particular, M does not accept as a sentence any string that is not generated by G. (This restriction can, of course, be softened by introducing degrees of grammaticalness, after the manner of Sec. 1.5, but we shall not burden the present discussion with that additional complication.) M is thus a finite transducer in the sense of Chapter 12, Sec. 1.5. It uses its information concerning the set of all strings in order to determine which of them are sentences of the language it understands and to understand sentences belonging to this language. This information, we assume, is represented in

the form of rules of the generative grammar G stored in the memory of M.

Before continuing, we should like to say once more that it is perfectly possible that M will not contain enough computing space to allow it to understand all sentences in the manner of the device G whose instructions it stores. This is no more surprising than the fact that a person who knows the rules of arithmetic perfectly may not be able to perform many computations correctly in his head. One must be careful not to obscure the fundamental difference between, on the one hand, a device M storing the rules G but having enough computing space to understand in the manner of G only a certain proper subset L' of the set L of sentences generated by G and, on the other hand, a device M^* designed specifically to understand only the sentences of L' in the manner of G. The distinction is perfectly analogous to the distinction between a device F that contains the rules of arithmetic but has enough computing space to handle only a proper subset Σ' of the set Σ of arithmetical computations and a device F^* that is designed to compute only Σ'. Thus, although identical in their behavior to F^* and M^*, F and M can improve their behavior without additional instruction if given additional memory aids, but F^* and M^* must be redesigned to extend the class of cases that they can handle. It is clear that F and M, the devices that incorporate competence whether or not it is realized in performance, provide the only models of any psychological relevance, since only they can explain the transfer of learning that we know occurs when memory aids are in fact made available.

In particular, if the grammar G incorporated in M exceeds any finite automaton in generative capacity, then we know that M will not be able to understand all sentences in the manner of G. There would be little reason to expect, a priori, that the natural languages learned by humans should belong to the special family of sets that can be generated by one-sided linear grammars (cf. Defs. 6 and 7, Chapter 12, Sec. 4.1) or by nonself-embedding context-free grammars (cf. Proposition 58 and Theorem 33, Chapter 12, Sec. 4.6). In fact, they do not, as we have observed several times. Consequently, we know that a realistic model M for the perceiver will incorporate a grammar G that generates sentences that M cannot understand in the manner of G (without additional aids). This conclusion should occasion no surprise; it leads to none of the paradoxical consequences that have occasionally been suggested. There has been much confusion about this matter and we should like to reemphasize the fact that the conclusion we have reached is just what should have been expected.

We can construct a model for the listener who understands a presented sentence by specifying the stored grammar G, the organization of memory, and the operations performable by M. We determine a class of perceptual models by stating conditions that these specifications must meet. In

Sec. 2.1 we consider perceptual models that store rewriting systems. Then in Sec. 2.2 we discuss possible features of perceptual models that incorporate transformational grammars.

2.1 Models Incorporating Rewriting Systems

Let us suppose that we have a language L generated by a context-sensitive grammar G that assigns to each sentence of L a P-marker—a labeled tree or labeled bracketing—in the manner we have already considered. What can we say about the understanding of sentences by the speaker of L? For example, what can we say about the class of sentences of his language that this speaker will be able to understand at all? If we construct a finite perceptual device M that incorporates the rules of G in its memory, to what extent will M be able to understand sentences in the manner of G?

In part, we answered this question in Sec. 4.6 of Chapter 12. Roughly, the answer was the following. Suppose that we say that *the degree of self-embedding of the P-marker Q is m* if m is the largest integer meeting the following condition: there is, in the labeled tree that represents Q, a continuous path passing through $m + 1$ nodes $N_0, \ldots, N_m$, each with the same label, where each N_i ($i \geqslant 1$) is fully self-embedded (with something to the left and something to the right) in the subtree dominated by N_{i-1}; that is to say, the terminal string of Q can be written in the form

$$xy_0y_1 \ldots y_{m-1}zv_{m-1} \ldots v_1v_0w, \tag{33}$$

where N_m dominates z, and for each $i < m$, N_i dominates

$$y_i \ldots y_{m-1}zv_{m-1} \ldots v_i, \tag{34}$$

and none of the strings $y_0, \ldots, y_{m-1}$, $v_0, \ldots, v_{m-1}$ is null. Thus, for example, in Fig. 5 the degree of self-embedding is two.

In Sec. 4.6 of Chapter 12 we presented a mechanical procedure Ψ that can be regarded as having the following effect: given a grammar G and an integer m, $\Psi(G, m)$ is a finite transducer M that takes a sentence x as input and gives as output a structural description $F(x)$ (which is, furthermore, a structural description assigned to x by G) wherever $F(x)$ has a degree of self-embedding of no more than m; that is to say, where m is a measure of the computing space available to a perceptual model M, which incorporates the grammar G, M will partially understand sentences in the manner of G just to the extent that the degree of self-embedding of their structural descriptions is not too great. As the amount of computing

space available to the device M increases, M will understand more deeply embedded structures in the manner of G. For any given sentence x there is an m sufficiently large so that the device M with computing space determined by m [i.e., the device $\Psi(G, m)$] will be capable of understanding x in the manner of G; M does not have to be redesigned to extend its capacities in this way. Furthermore, this is the best result that can be achieved, since self-embedding is, as was proved in Chapter 12, precisely the property that distinguishes context-free languages from the regular languages that can be generated (accepted) by finite automata.

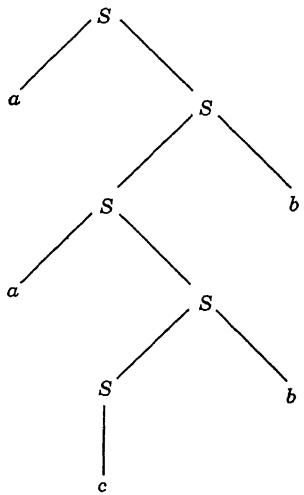


Fig. 5. Phrase marker with a degree of self-embedding equal to two.

In Chapter 12 this result was stated only for a certain class K of context-free grammars. We pointed out that the class K contains a grammar for every context-free language and that it is a straightforward matter to drop many, if not all, of the restrictions that define K. Extension to context-sensitive grammars is another matter, however, and the problem of finding an optimal finite transducer that understands the sentences of G as well as possible, for any context-sensitive G, has not been investigated at all. Certain approaches to this question are suggested by the results of Matthews', discussed in Chapter 12, Sec. 4.2, on asymmetrical context-sensitive grammars and PDS automata, but these have not yet been pursued.

These restrictions aside, the procedure Ψ of Sec. 4.6, Chapter 12, provides an optimal perceptual model (i.e., an optimal finite recognition routine) that incorporates a context-free grammar G. Given G, we can immediately construct such a device in a mechanical way, and we know that it will do as well as can be done by any device with bounded memory in understanding sentences in the manner of G. As the amount of memory increases, its capacity to understand sentences of G increases without limit. Only self-embedding beyond a certain degree causes it to fail when memory is fixed. We can, in fact, rephrase the construction so that the procedure Ψ determines a transducer $\Psi(G)$ which understands all sentences in the manner of G, where $\Psi(G)$ is a "single-pass" device with only push-down storage, as shown in Sec. 4.2, Chapter 12.

Observe that the optimal perceptual model $M = \Psi(G, m)$, where m is fixed, may fail to understand sentences in the manner of G even when

the language L generated by G might have been generated by a one-sided linear grammar (finite automaton). For example, the context-free grammar G that gives the structural description in Fig. 5 might be the following:

$$S \to aS, \qquad S \to Sb, \qquad S \to c. \tag{35}$$

(It is a straightforward matter to extend Ψ to deal with rules of the kind in Example 35.) The generated language is the set of all strings $a^i cb^j$ and is clearly a regular language. Nevertheless, with $m = 1$, $\Psi(G, m)$ will not be capable of understanding the sentence *aacbb* generated in Fig. 5 *in the manner of* G, since this derivation has a degree of self-embedding equal to two. The point is that although a finite automaton can be found to accept the sentences of this language it is not possible to find a finite device that understands all of its sentences in the manner of the particular generative process G represented in Example 35.

Observe also that the perceptual device $\Psi(G, m)$ is nondeterministic. As a perceptual model it has the following defect. Suppose that G assigns to x a structural description D with degree of self-embedding not exceeding m. Then, as we have indicated, the device $\Psi(G, m)$ will be capable of computing in such a way that it will map x into D, thus interpreting x in the manner of G. Being nondeterministic, however, it may also, given x, compute in such a way that it will fail to map x into a structural description at all. If $\Psi(G, m)$ fails to interpret x in the manner of G on a particular computation, we can conclude nothing about the status of x with respect to the grammar G, although if $\Psi(G, m)$ does map x into a structural description D we can conclude that G assigns D to x. We might investigate the problem of constructing a deterministic perceptual model that partially understands the output of a context-free grammar, or a model with nondeterminacy matching the ambiguity of the underlying grammar—that is, a model that may block on a computation with a particular string only if this string is either not generated by the grammar from which the model is constructed or is generated only by a derivation that is too deeply self-embedded for the device in question—but this matter has not yet been carefully investigated. It is clear, however, that such devices unlike $\Psi(G, m)$, would involve a restriction on the right-recursive elements in the structural descriptions (i.e., on right branchings). See, in this connection, the example on p. 473.

Self-embedding is the fundamental property that takes a system outside of the generative capacity of a finite device, and self-embedding will ultimately result from nesting of dependencies, since the nonterminal vocabulary is finite. However, the nesting of dependencies, even short of self-embedding, causes the number of states needed in the device $\Psi(G, m)$ to

increase quite rapidly with the length of the input string that it is to understand. Consequently, we would expect that nested constructions should become difficult to understand even when they are, in principle, within the capacity of a finite device, since available memory (i.e., number of states) is clearly quite limited for real-time analytic operations, a fact to which we return in Sec. 2.2. Indeed, as we observed in Chapter 11 (cf. Example 11 in Sec. 3), nested structures even without self-embedding quickly become difficult or impossible to understand.

From these observations we are led to conclude that sentences of natural languages containing nested dependencies or self-embedding beyond a certain point should be impossible for (unaided) native speakers to understand. This is indeed the case, as we have already pointed out. There are many syntactic devices available in English—and in every other language that has been studied from this point of view—for the construction of sentences with nested dependencies. These devices, if permitted to operate freely, will quickly generate sentences that exceed the perceptual capacities (i.e., in this case, the short-term memory) of the native speakers of the language. This possibility causes no difficulties for communication, however. These sentences, being equally difficult for speaker and hearer, simply are not used, just as many other proliferations of syntactic devices that produce well-formed sentences will never actually be found.

There would be no reason to expect that these devices (which are, of course, continually used when nesting is kept within the bounds of memory restriction) should disappear as the language evolves; and, in fact, they do not disappear, as we have observed. It would be reasonable to expect, however, that a natural language might develop techniques to paraphrase complex nested sentences as sentences with either left-recursive or right-recursive elements, so that sentences of the same content could be produced with less strain on memory. That expectation, formulated by Yngve (1960, 1961) in a rather different way, to which we return, is well confirmed. Alongside such self-embedding English sentences as *if, whenever X then Y, then Z*, we can have the basically right-branching structure *Z if whenever X, then Y*, and so on in many other cases. In particular, many singulary grammatical transformations in English seem to be primarily stylistic; they convert one sentence into another with much the same content but with less self-embedding. Alongside the sentence *that the fact that he left was unfortunate is obvious*, which doubly embeds *S*, we have the more intelligible and primarily right-recursive structure *it is obvious that it was unfortunate that he left*. Similarly, we have a transformation that converts *the cover that the book that John has has* to *John's book's cover*, which is left-branching rather than self-embedding. (It should also be noted, however, that some of these so-called stylistic transformations can increase

structural complexity, e.g., those that give "cleft-sentences"—from *I read the book that you told me about* we can form *it was the book that you told me about that I read*, etc.)

Now to recapitulate: from the fact that human memory is finite we can conclude only that some self-embedded structures should not be understandable; from the further assumption that memory is small, we can predict difficulties even with nested constructions. Although sentences are accepted (heard and spoken) in a single pass from left to right, we cannot conclude that there should be any left-right asymmetry in the understandable structures. Nor is there any evidence presently available for such asymmetry. We have little difficulty in understanding such right-branching constructions as *he watched the boy catch the ball that dropped from the tower near the lake* or such left-branching constructions as *all of the men whom I told you about who were exposed to radiation who worked half-time are still healthy, but the ones who worked full time are not* or *many more than half of the rather obviously much too easily solved problems were dropped last year*. Similarly, no conclusion can be drawn from our present knowledge of the distribution of left-recursive and right-recursive elements in language. Thus, in English, right-branching constructions predominate; in other languages—Japanese, Turkish—the opposite is the case. In fact, in every known language we find right-recursive, left-recursive, and self-embedding elements (and, furthermore, we find coordinate constructions that exceed the capacity of rewriting systems entirely, a fact to which we return directly).

We have so far made only the following assumptions about the model M for the user:

1. M is finite;
2. M accepts (or produces) sentences from left-to-right in a single pass;
3. M incorporates a context-free grammar as a representation of its competence in and knowledge of the language.

Of these, (3) is surely false, but the conclusions concerning recursive elements that we have drawn from it would undoubtedly remain true under a wide class of more general assumptions. Obviously, (1) is beyond question; (2) is an extremely weak assumption that also cannot be questioned, either for the speaker or hearer—note that many different kinds of internal organization of M are compatible with (2), for example, the assumption that M stores a finite string before deciding on the analysis of its first element or that M stores a finite number of alternative assumptions about the first element which are resolved only at an indefinitely later time.

If we add further assumptions beyond these three, we can derive additional conclusions about the ability of the device to produce or understand sentences in the manner of the incorporated grammar. Consider the two extreme assumptions:

4. M produces P-markers strictly "from the top down," or from trunk to branch, in the tree graph of the P-marker.

5. M produces P-markers strictly "from the bottom up," or from branch to trunk, in the tree graph of the P-marker.

In accordance with (4), the device M will interpret a rule $A \rightarrow \phi$ of the incorporated grammar as the instruction "rewrite A as ϕ"—that is to say, as the instruction that, in constructing a derivation, a line of the form $\psi_1 A \psi_2$ can be followed by the line $\psi_1 \phi \psi_2$. Assumption 5 requires the device M to interpret each rule $A \rightarrow \phi$ of the grammar as the instruction "replace ϕ by A"—that is to say, in constructing an inverted derivation with S as its last line and a terminal string as its first line, a line of the form $\psi_1 \phi \psi_2$ can be followed by the line $\psi_1 A \psi_2$.

From Assumption 4 we can conclude that only a bounded number of successive left-branchings can, in general, be tolerated by M. Thus suppose that M is based on a grammar containing the rule $S \rightarrow SA$. After n applications of this left-branching rule the memory of a device meeting Assumptions 2 and 4 (under the natural interpretation) would have to store n occurrences of A for later rewriting and would thus eventually have to violate Assumption 1. On the other hand, from Assumption 5 we can conclude that only a bounded number of successive right-branchings can in general be tolerated. For example, suppose the underlying grammar contains right-branching rules: $A \rightarrow cA$, $B \rightarrow cB$, $A \rightarrow a$, and $B \rightarrow b$. In this case the device will be presented with strings $c^n a$ or $c^n b$. Now, although Assumption 2 still calls for resolution from left to right, Assumption 5 implies that no node in the P-marker can be replaced until all that it dominates is known, so that resolution must be postponed until the final symbol in the string is received. Thus the device would have to store n occurrences of c for later rewriting and, again, Assumption 1 must eventually be violated. Left-branching causes no difficulty under Assumption 5, of course, just as right-branching causes no difficulty in the case of Assumption 4. Thus Assumptions 4 and 5 impose left-right asymmetries (in opposite ways) on the set of structures that can be accepted or produced by M. Observe that the devices $\Psi(G, m)$, given by the procedure Ψ of Chapter 12, Sec. 4.6, need not meet either of the restrictions in Assumption 4 or 5; in constructing a particular P-marker, they may move up or down or both ways indefinitely often, just as long as self-embedding is restricted.

Assumption 4 might be interpreted as a condition to be met by the speaker; Assumption 5, as a condition to be met by the hearer. (Of course, if we design a model of the speaker to meet Assumption 4 and a model of the hearer to meet Assumption 5 simultaneously, we will severely restrict the possibility of communication between them.) If Assumption 4 described the speaker, we would expect him to have difficulty with left-branching constructions; if Assumption 5 described the listener, we would expect him to have difficulty with right-branching constructions. Neither assumption seems particularly plausible. There is no reason to think that a speaker must always select his major phrase types before the minor subphrases or his word categories before his words (Assumption 4). Similarly, although a listener obviously receives terminal symbols and constructs phrase types, there is no reason to assume that decisions concerning minor phrase types must uniformly precede those concerning major structural features of the sentence. Assumptions 4 and 5 are but two of a large set of possible assumptions that might be considered in specifying models of the user more fully. Thus we might introduce an assumption that there is a bound on the length of the string that must be received before a construction can be uniquely identified by a left-to-right perceptual model—and so on, in many other ways.

There has been some discussion of hypotheses such as Assumptions 4 and 5. For example, Skinner's (1957) proposal that "verbal operant responses" to situations (e.g., the primary nouns, verbs, adjectives) form the raw materials of which sentences are constructed by higher level "autoclitic" responses (grammatical devices, ordering, selecting, etc.) might be loosely interpreted as a variant of Assumption 5, regarded as an assumption about the speaker. Yngve (1960, 1961) has proposed a variant of (4) as an assumption about the speaker; his proposal is explicitly directed toward our present topic and so demands a somewhat fuller discussion.

Yngve describes a process by which a device that contains a grammar rather similar to a context-free grammar produces derivations of utterances, always rewriting the leftmost nonterminal symbol in the last line of the already constructed derivation and postponing any nonterminal symbols to the right of it. Each postponed symbol, therefore, is a promise that must be remembered until the time comes to develop it; as the number of these promises grows, the load on memory also grows. Thus Yngve defines a measure of *depth* in terms of the number of postponed symbols, so that left-branching, self-embedding, and multiple-branching all contribute to depth, whereas right-branching does not. (Note that the depth of postponed symbols and the degree of embedding are quite distinct measures.) Yngve observes that a model so constructed for the speaker

will be able with a limited memory to produce structures that do not exceed a certain depth. He offers the hypothesis that Assumption 4, so interpreted, is a correct characterization of the speaker and that natural languages have developed in such a way as to ease the speaker's task by limiting the necessity for left-branching.

The arguments in support of this hypothesis, however, seem inconclusive. It is difficult to see why any language should be designed for the ease of the speaker rather than the hearer, and Assumption 4 in any form seems totally unmotivated as a requirement for the hearer; on the contrary, the opposite assumption, as we have noted, seems the better motivated of the two. Nor does (4) seem to be a particularly plausible assumption concerning the speaker, for reasons we have already stated. It is possible, of course, to construct sentences that have a great depth and that are quite unintelligible, but they characteristically involve nesting or self-embedding and thus serve merely to show that the speaker and hearer have finite memories—that is to say, they support only the obvious and unquestionable Assumptions 1 and 2, not the additional Assumption 4. In order to support Yngve's hypothesis, we would have to find unintelligible sentences whose difficulty was attributable entirely to left-branching and multiple-branching. Such examples are not readily produced. In order to explain why multiple-branching, which contributes to the measure of depth, does not cause more difficulty, Yngve treats coordinate constructions (e.g., conjunctions) as right-branching, which does not contribute to the number of postponed symbols. But this is perfectly arbitrary; they could just as well be treated as left-branching. The only correct interpretation for such constructions is in terms of multiple-branching from a single node—this is exactly the formal feature that distinguishes true coordinate constructions, with no internal structure, from others. As we have observed in Chapter 11, Sec. 5, such constructions are beyond the limits of systems of rewriting rules altogether. Hence the relative ease with which such sentences as Examples 18 and 20 of Chapter 11 can be understood contradicts not only Assumption 4 but even the underlying Assumption 3, of which 4 is an elaboration.

In short, there seems to be little that we can say about the speaker and the hearer beyond the obvious fact that they are limited finite devices that relate sentences and structural descriptions and that they are subject to the constraint that time is linear. From this, all that we can conclude is that self-embedding (and, more generally, nesting of dependencies) should cause difficulty, as indeed it does. It is also not without interest that self-embedding seems to impose a greater burden than an equivalent amount of nesting without self-embedding. Further speculations are, at the present time, quite unsupported.

2.2 Models Incorporating Transformational Grammars

There are surprising limitations on the amount of short-term memory available for human data processing, although the amount of long-term memory is clearly great (cf. Miller, 1956). This fact suggests that it might be useful to look into the properties of a perceptual model M with two basic components, M_1 and M_2, operating as follows: M_1 contains a small, short-term memory. It performs computations on an input string x as it is received symbol by symbol and transmits the result of these computations to M_2. M_2 contains a large long-term memory in which is stored a generative grammar G; the task of M_2 is to determine the deeper structure of the input string x, using as its information the output transmitted to it by M_1. (Sentence-analyzing procedures of this sort have been investigated by Matthews, 1961.)

The details of the operation of M_2 would be complicated, of course; probably the best way to get an appreciation of the functions it would have to perform is to consider an example in some detail. Suppose, therefore, that a device M, so constructed, attempts to analyze such sentences as

John is easy to please. (36)

John is eager to please. (37)

To these, M_1 might assign preliminary analyses, as in Fig. 6, in which inessentials are omitted. Clearly, however, this is not the whole story. In order to account for the way in which we understand these sentences, it is necessary for the component M_2, accepting the analysis shown in Fig. 6 as input, to give as output structural descriptions that indicate that in

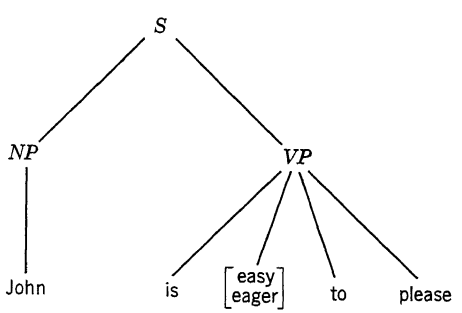


Fig. 6. Preliminary analysis of Sentences 36 and 37.

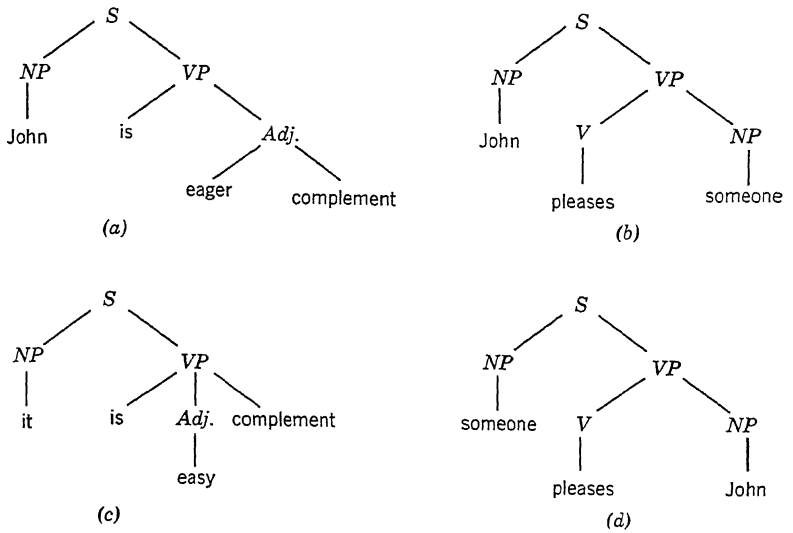


Fig. 7. Some *P*-markers that would be generated by the rewriting rules of the grammar and to which the transformation rules would apply.

Example 36 *John* is the direct object of *please*, whereas in Example 37 it is the logical subject of *please*.

Before we can attempt to provide a description of the device M_2 we must ask how structural information of this deeper kind can be represented. Clearly, it cannot be conveyed in the labeled tree (*P*-marker) associated with the sentence as it stands. No elaboration of the analysis shown in Fig. 6, with more elaborate subcategorization, etc., will remedy the fundamental inability of this form of representation to mirror grammatical relations properly. We are, of course, facing now precisely the kind of difficulty that was discussed in Chapter 11, Sec. 5, and that led to the development of a theory of transformational generative grammar. In a transformational grammar for English the rewriting rules would not be required to provide Examples 36 and 37 directly; the rewriting rules would be limited to the generation of such *P*-markers as those shown in Fig. 7 (where inessentials are again omitted). In addition, the grammar will contain such transformations as

T_1: replaces *complement* by "for x to y," where x is an *NP* and y is a *VP* in the already generated sentence xy;
T_2: deletes the second occurrence of two identical *NP*'s (with whatever is affixed to them);
T_3: deletes direct objects of certain verbs;

T_4: deletes "for someone" in certain contexts;
T_5: converts a string analyzable as

$$NP - \text{is} - Adj - (\text{for} - NP_1) - \text{to} - V - NP_2$$

to the corresponding string of the form

$$NP_2 - \text{is} - Adj - (\text{for} - NP_1) - \text{to} - V.$$

Each of these can be generalized and put in the form specified in Chapter 11. When appropriately generalized, they are each independently motivated by examples of many other kinds. Note, for example, the range of sentences that are similar in their underlying structural features to Examples 36 and 37; we have such sentences as *John is an easy person to please*, *John is a person who (it) is easy to please*, *this room is not easy to work in (to do decent work in)*, *he is easy to do business with*, *he is not easy to get information from*, *such claims are very easy to be fooled by*, and many others all of which are generated in essentially the same way.

Applying T_1 to the pair of structures in Figs. 7*c* and 7*d*, we derive the sentence *It is easy for someone to please John*, with its derived *P*-marker. Applying T_4 to this, we derive *It is easy to please John*, which is converted to Example 36 by T_5. Had we applied T_5 without T_4, we could have derived, for example, *John is easy for us to please* (with *we* chosen in place of *someone* in Fig. 7*d*—we leave unstated obvious obligatory rules). Applying T_1 to the pair of structures in Figs. 7*a* and 7*b*, we derive *John is eager for John to please someone*, which is converted by T_2 to *John is eager to please someone*. Had we applied T_3 to Fig. 7*b* before applying T_1, we would, in the same way, have derived Example 37.

At this point we should comment briefly on several features of such an analysis. Notice that *I am eager for you to please*, *you are eager for me to please*, etc., are all well-formed sentences; but *I am eager for me to please*, *you are eager for you to please*, etc., are impossible and are reduced to *I am eager to please*, *you are eager to please* obligatorily by T_2. This same transformation gives *I expected to come*, *you expected to come*, etc., from *I expected me to come*, *you expected you to come*, which are formed in the same way as *you expected me to come*, *I expected you to come*. Thus this grammar does actually regard *John* in Example 37 as identical with the deleted subject of *please*. Note, in fact, that in the sentence *John expected John to please*, in which T_2 has *not* applied, the two occurrences of *John* must have different reference. In Example 36, on the other hand, *John* is actually the direct object of *please*, assuming grammatical relations to be preserved under transformation (assuming, in other words, that the *P*-marker represented in Fig. 7*d* is part of the structural description of

Example 36). Note, incidentally, that T_5 does not produce such non-sentences as *John is easy to come*, since there is no *NP comes John*, though we have *John is eager to come* by T_1, T_2. T_5 would not apply to any sentence of the form

$$NP - \text{is} - \text{eager} - (\text{for} - NP_1) - \text{to} - V - NP_2$$

to give

$$NP_2 - \text{is} - \text{eager} - (\text{for} - NP_1) - \text{to} - V$$

(for example, *Bill is eager for us to meet* from *John is eager for us to meet Bill*; *these crooks are eager for us to vote out* from *John is eager for us to vote out these crooks*), since *eager complement*, but not *eager*, is an *Adj* (whereas, *easy*, but not *easy complement*, is an *Adj*). Supporting this analysis is the fact that the general rule that nominalizes sentences of the form *NP — is — Adj* (giving, for example, *John's cleverness* from *John is clever*), converts *John is eager (for us) to come* (which comes from Fig. 7*a* and *we come* by T_1) to *John's eagerness for us to come*; but it does not convert Example 36 to *John's easiness to please*. Furthermore, the general transformational process that converts phrases of the form

$$\text{the} - \text{Noun} - \text{who (which)} - \text{is} - Adj$$

to

$$\text{the} - Adj - \text{Noun}$$

(for example, *the man who is old* to *the old man*) does convert *a fellow who is easy to please* to *an easy fellow to please* (since *easy* is an *Adj*) but does not convert *a fellow who is eager to please* to *an eager fellow to please* (since *eager* is not, in this case, an *Adj*). In brief, when these rules are stated carefully, we find that a large variety of structures is generated by quite general, simple, and independently motivated rules, whereas other superficially similar structures are correctly excluded. It would not be possible to achieve the same degree of generalization and descriptive adequacy with a grammar that operates in the manner of a rewriting system, assigning just a single *P*-marker to a sentence as its structural description.

Returning now to our main theme, we see that the grammatical relations of *John* to *please* in Examples 36 and 37 are represented in the intuitively correct way in the structural descriptions provided by a transformational grammar. The structural description of Example 36 consists of the two underlying *P*-markers in Figs. 7*c* and 7*d* and the derived *P*-marker in Fig. 6 (as well as a record of the transformational history, i.e., T_1, T_4, T_5). The structural description of Example 36 consists of the underlying *P*-markers in Figs. 7*a* and 7*b* and the derived *P*-marker in Fig. 6 (along with the transformational history T_1, T_2, T_3). Thus the structural description

of Example 36 contains the information that *John* in Example 36 is the object of *please* in the underlying *P*-marker of Fig. 7*d*; and the structural description of Example 37 contains the information that *John* in Example 37 is the subject of *please* in the underlying *P*-marker in Fig. 7*b*. Note that, when the appropriately generalized form of T_5 applies to *it is easy to do business with John* to yield *John is easy to do business with*, we again have in the underlying *P*-markers a correct account of the grammatical relations in the transform, although in this case the grammatical subject *John* is no longer the object of the verb of the complement, as it is in Example 3*b*. Notice also that it is the underlying *P*-markers, rather than the derived *P*-marker, that represent the semantically relevant information in this case. In this respect, these examples are quite typical of what is found in more extensive grammars.

These observations suggest that the transformational grammar be stored and utilized only by the component M_2 of the perceptual model. M_1 will take a sentence as input and give us as output a relatively superficial analysis of it (perhaps a derived *P*-marker such as that in Fig. 6). M_2 will utilize the full resources of the transformational grammar to provide a structural description, consisting of a set of *P*-markers and a transformational history, in which deeper grammatical relations and other structural information are represented. The output of $M = (M_1, M_2)$ will be the complete structural description assigned to the input sentence by the grammar that it stores; but the analysis that is provided by the initial, short-term memory component M_1 may be extremely limited.

If the memory limitations on M_1 are severe, we can expect to find that structurally complex sentences are beyond its analytic power even when they lack the property (i.e., repeated self-embedding) that takes them completely beyond the range of any finite device. It might be useful, therefore, to develop measures of various sorts to be correlated with understandability. One rough measure of structural complexity that we might use, along with degree of nesting and self-embedding, is the node-to-terminal-node ratio $N(Q)$ in the *P*-marker Q of the terminal string $t(Q)$. This number measures roughly the amount of computation per input symbol that must be performed by the listener. Hence an increase in $N(Q)$ should cause a correlated difficulty in interpreting $t(Q)$ for a real-time device with a small memory. Clearly $N(Q)$ grows as the amount of branching per node decreases. Thus $N(Q)$ is higher for a binary *P*-marker such as that shown in Fig. 8*a* than for the *P*-marker in Fig. 8*b* that represents a coordinate construction with the same number of terminals. Combined with our earlier speculations concerning the perceptual model M, this observation would lead us to suspect that $N(Q)$ should in general be higher for the derived *P*-marker that must be provided by the limited

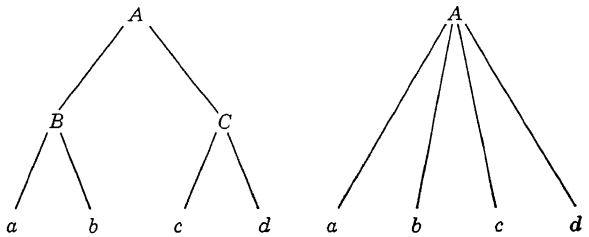


Fig. 8. Illustrating a measure of structural complexity. $N(Q)$ for the P-marker (a) is 7/4; for (b), $N(Q) = 5/4$.

component M_1 than it would be for underlying P-markers. In other words, the general effect of transformations should be to decrease the total amount of structure in the associated P-marker. This expectation is fully borne out. The underlying P-markers have limited, generally binary branching. But, as we have already observed in Chapter 11 (particularly p. 305), binary branching is not a general characteristic of the derived P-markers associated with actual sentences; in fact, the actual set of derived P-markers is beyond the generative capacity of rewriting systems altogether, since there is no bound on the amount of branching from a single node (that is to say, on the length of a coordinate construction).

The psychological plausibility of a transformational model of the language user would be strengthened, of course, if it could be shown that our performance on tasks requiring an appreciation of the structure of transformed sentences is some function of the nature, number, and complexity of the grammatical transformations involved.

One source of psychological evidence concerns the grammatical transformation that negates an affirmative sentence. It is a well-established fact that people in concept-attainment experiments find it difficult to use negative instances (Smoke, 1933). Hovland and Weiss (1953) established that this difficulty persists even when the amount of information conveyed by the negative instances is carefully equated to the amount conveyed by positive instances. Moreover, Wason (1959, 1961) has shown that the grammatical difference between affirmative and negative English sentences causes more difficulty for subjects than the logical difference between true and false; that is to say, if people are asked to verify or to construct simple sentences (about whether digits in the range 2 to 9 are even or odd), they will take longer and make more errors on the true negative and false negative sentences than on the true affirmative and false affirmative sentences. Thus there is some reason to think that there may be a grammatical explanation for some of the difficulty we have in using negative information; moreover, this speculation has received some support from

Eifermann (1961), who found that negation in Hebrew has a somewhat different effect on thinking than it has in English.

A different approach can be illustrated by sentence-matching tests (Miller, 1962*b*). One study used a set of 18 elementary strings (for example, those formed by taking *Jane*, *Joe*, or *John* as the first constituent, *liked* or *warned* as the second, and *the old woman*, *the small boy*, or *the young man* as the last), along with the corresponding sets of sentences that could be formed from those by passive, negative, or passive-and-negative transformations. These sets were taken two at a time, and subjects were asked to match the sentences in one set with the corresponding sentences in the other. The rate at which they worked was recorded and from that it was possible to obtain an estimate of the time required to perform the necessary transformations. If we assume that these four types of sentence are coordinate and independently learned, then there is little reason to believe that finding correspondences between any two of them will necessarily be more difficult than between any other two. On the other hand, if we assume that the four types of sentence are related to one another by two grammatical transformations (and their inverses), then we would expect some of the tests to be much easier than others. The data supported a transformational position: the negative transformation was performed most rapidly, the more complicated passive transformation took slightly longer, and tests requiring both transformations (kernel to passive-negative or negative to passive) took as much time as the two single transformations did added together. For example, in order to perform the transformations necessary to match such pairs as *Jane didn't warn the small boy* and *The small boy was warned by Jane*, subjects required on the average more than three seconds, under the conditions of the test.

Still another way to explore these matters is to require subjects to memorize a set of sentences having various syntactic structures (J. Mehler, personal communication). Suppose, for example, that a person reads at a rapid but steady rate the following string of eight sentences formed by applying passive, negative, and interrogative transformations: *Has the train hit the car? The passenger hasn't been carried by the airplane. The photograph has been made by the boy. Hasn't the girl worn the jewel? The student hasn't written the essay. The typist has copied the paper. Hasn't the house been bought by the man? Has the discovery been made by the biologist?* When he finishes, he attempts to write down as many as he can recall. Then the list (in scrambled order) is read again, and again he tries to recall, and so on through a series of trials. Under those conditions many syntactic confusions occur, but most of them involve only a single transformational step. It is as if the person recoded the original sentences

into something resembling a kernel string plus some correction terms for the transformations that indicate how to reconstruct the correct sentence when he is called on to recite. During recall he may remember the kernel, but become confused about which transformations to apply.

Preliminary evidence from these and similar studies seems to support the notion that kernel sentences play a central role, not only linguistically, but psychologically as well. It also seems likely that evidence bearing on the psychological reality of transformational grammar will come from careful studies of the genesis of language in infants, but we shall not attempt to survey that possibility here.

It should be obvious that the topics considered in this section have barely been opened for discussion. The problem can clearly profit from abstract study of various kinds of perceptual models that incorporate generative processes as a fundamental component. It would be instructive to study more carefully the kinds of structures that are actually found in natural languages and the formal features of those structures that make understanding and production of speech difficult. In this area the empirical study of language and the formal study of mathematical models may bear directly on questions of immediate psychological interest in what could turn out to be a highly fruitful and stimulating way.

3. TOWARD A THEORY OF COMPLICATED BEHAVIOR

It should by now be apparent that only a complicated organism can exploit the advantages of symbolic organization. Subjectively, we seem to grasp meanings as integrated wholes, yet it is not often that we can express a whole thought by a single sound or a single word. Before they can be communicated, ideas must be analyzed and represented by sequences of symbols. To map the simultaneous complexities of thought into a sequential flow of language requires an organism with considerable power and subtlety to symbolize and process information. These complexities make linguistic theory a difficult subject. But there is an extra reward to be gained from working it through. If we are able to understand something about the nature of human language, the same concepts and methods should help us to understand other kinds of complicated behavior as well.

Let us accept as an instance of complicated behavior any performance in which the behavioral sequence must be internally organized and guided by some hierarchical structure that plays the same role, more or less, as a *P*-marker plays in the organization of a grammatical sentence. It is not

immediately obvious, of course, how we are to decide whether some particular nonlinguistic performance is complicated or simple; one natural criterion might be the ability to interrupt one part of the performance until some other part had been completed.

The necessity for analyzing a complex idea into its component parts has long been obvious. Less obvious, however, is the implication that any complicated activity obliges us to analyze and to postpone some parts while others are being performed. A task, X, say, is analyzed into the parts Y_1, Y_2, Y_3, which should, let us assume, be performed in that order. So Y_1 is singled out for attention while Y_2 and Y_3 are postponed. In order to accomplish Y_1, however, we find that we must analyze it into Z_1 and Z_2, and those in turn must be analyzed into still more detailed parts. This general situation can be expressed in various ways—by an outline or by a list structure (Newell, Shaw, & Simon, 1959) or by a tree graph similar to those used to summarize the structural description of individual sentences. While one part of a total enterprise is being accomplished, other parts may remain implicit and still largely unformulated. The ability to remember the postponed parts and to return to them in an appropriate order is necessarily reserved for organisms capable of complicated information processing. Thus the kind of theorizing we have been doing for sentences can easily be generalized to even larger units of behavior. Restricted-infinite automata in general, and PDS systems in particular, seem especially appropriate for the characterization of many different forms of complicated behavior.

The spectrum of complicated behavior extends from the simplest responses at one extreme to our most intricate symbolic processes at the other. In gross terms it is apparent that there is some scale of possibilities between these extremes, but exactly how we should measure it is a difficult problem. If we are willing to borrow from our linguistic analysis, there are several measures already available. We can list them briefly:

INFORMATION AND REDUNDANCY. The variety and stereotypy of the behavior sequences available to an organism are an obvious parameter to estimate in considering the complexity of its behavior (cf. Miller & Frick, 1949; Frick & Miller, 1951).

DEGREE OF SELF-EMBEDDING. This measure assumes a degree of complication that may seldom occur outside the realm of language and language-mediated behaviors. Self-embedding is of such great theoretical significance, however, that we should certainly look for occurrences of it in nonlinguistic contexts.

DEPTH OF POSTPONEMENT. This measure of memory load, proposed by Yngve, may be of particular importance in estimating a person's

capacity to carry out complicated instructions or consciously to devise complicated plans for himself.

STRUCTURAL COMPLEXITY. The ratio of the total number of nodes in the hierarchy to the number of terminal nodes provides an estimate of complexity that, unlike the depth measure, is not asymmetrical toward the future.

TRANSFORMATIONAL COMPLEXITY. A hierarchical organization of behavior to meet some new situation may be constructed by transforming an organization previously developed in some more familiar situation. The number of transformations involved would provide an obvious measure of the complexity of the transfer from the old to the new situation.

These are some of the measures that we can adapt in analogy to the linguistic studies; no doubt many others of a similar nature could be developed.

Clearly, no one can look at a single instance of some performance and immediately assign values to it for any of those measures. As in the case of probability measures, repeated observations under many different conditions are required before a meaningful estimate is available.

Many psychologists, of course, prefer to avoid complicated behavior in their experimental studies; as long as there was no adequate way to cope with it, the experimentalist had little other alternative. Since about 1945, however, this situation has been changing rapidly. From mathematics and logic have come theoretical studies that are increasingly suggestive, and the development of high-speed digital computers has supplied a tool for exploring hypotheses that would have seemed fantastic only a generation ago. Today, for example, it is becoming increasingly common for experimental psychologists to phrase their theories in terms of a computer program for simulating behavior (cf. Chapter 7). Once a theory is expressed in that form, of course, it is perfectly reasonable to try to apply to it some of the indices of complexity.

Miller, Galanter, and Pribram (1960) have discussed the organization of complicated behavior in terms of a hierarchy of tote units. A *tote unit* consists of two parts: a *test* to see if some situation matches an internally generated criterion and an *operation* that is intended to reduce any differences between the external situation and some internal criterion. The criterion may derive from a model or hypothesis about what will be perceived or what would constitute a satisfactory state of affairs. The operations can either revise the criterion in the light of new evidence received or they can lead to actions that change the organism's internal and/or external environment. The test and its associated operations are actively linked in a feedback loop to permit iterated adjustments until the criterion

is reached. A tote (test-operate-test-exit) unit is shown in the form of a flow-chart in Fig. 9. A hierarchy of tote units can be created by analyzing the operational phase into a sequence of tote units; then the operational phase of each is analyzed in turn. There should be no implication, however, that the hierarchy must be constructed exclusively from strategy to tactics or exclusively from tactics to strategy—both undoubtedly occur. An example of the kind of structures produced in this way is shown in the flowchart in Fig. 10.

These serial flowcharts are simply the finite automata we considered in Chapter 12, and it is convenient to replace them by oriented graphs (cf. Karp, 1960). Wherever an initial or terminal element or operation occurs in the flowchart, replace it by a node with one labeled arrow exiting from the node; wherever a test occurs, replace it by a node with two labeled exits. Next, replace every nonbranching sequence of arrows by a single arrow bearing a compound label. The graph corresponding to the flow-chart of Fig. 10 is shown in Fig. 11. From such oriented graphs as these it is a simple matter to read off the set of triples that define a finite automaton.

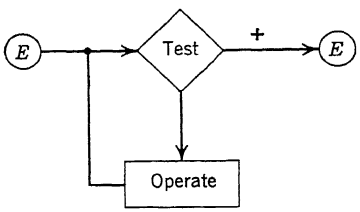


Fig. 9. A simple tote unit.

A tote hierarchy is just a general form of finite automaton in the sense of Chapter 12. We know from Theorem 2 of Chapter 12 that for any finite automaton there is an equivalent automaton that can be represented by a finite number of finite notations of the form $A_1(A_2, \ldots, A_m)^* A_{m+1}$, where the elements $A_2, \ldots, A_m$ can themselves be notations of the same form, and so on, until the full hierarchy is represented. For any finite state model that may be proposed, therefore, there is an equivalent model in terms of a (generalized) tote hierarchy.

Since a tote hierarchy is analogous to a program of instructions for a serial computer, it has been referred to as a *plan* that the system is trying to execute. Any postponed parts of the plan constitute the system's *intentions* at any given moment. Viewed in this way, therefore, the finite devices discussed in these chapters are clearly applicable to an even broader range of behavioral processes than language and communication. Some implications of this line of argument for nonlinguistic phenomena have been discussed informally by Miller, Galanter, and Pribram.

A central concern for this type of theory is to understand where new plans come from. Presumably, our richest source of new plans is our old plans, transformed to meet new situations. Although we know little about it, we must have ways to treat plans as objects that can be formed and transformed according to definite rules. The consideration of transformational grammars gives some indication of how we might combine

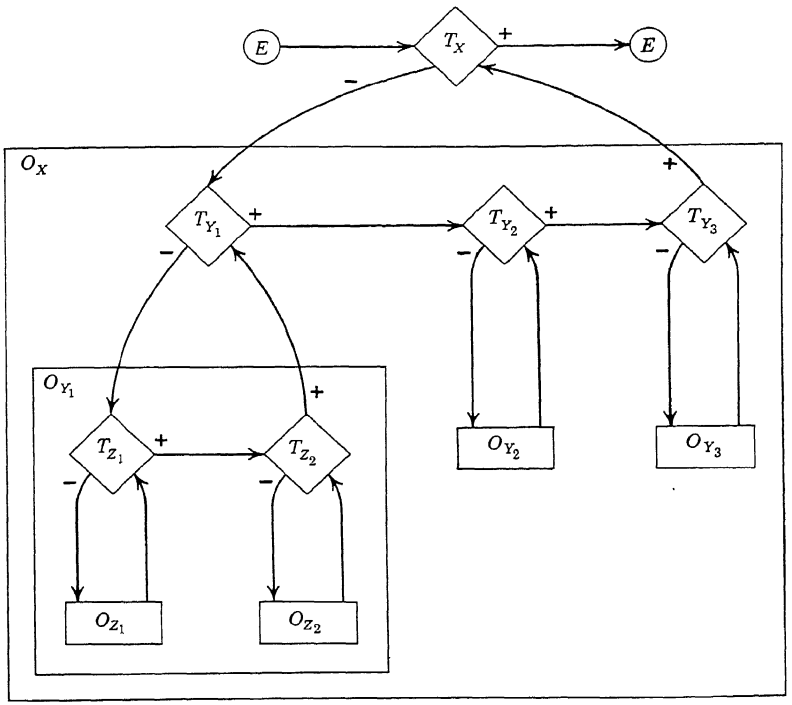


Fig. 10. A hierarchical system of tote units.

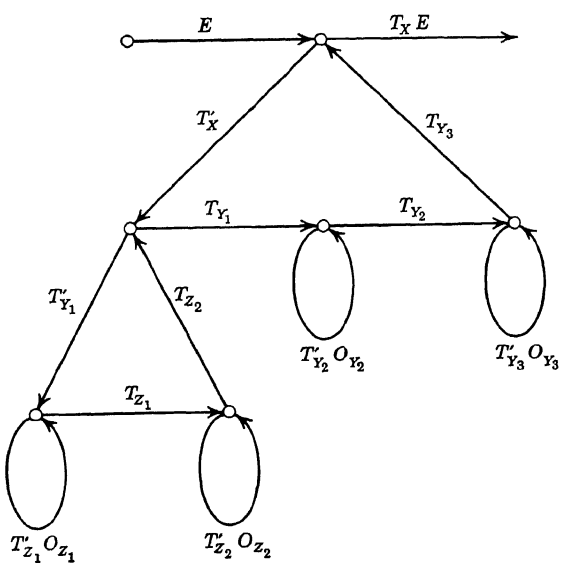


Fig. 11. Graph of flowchart in Fig. 10.

and rearrange plans, which are, of course, so closely analogous to *P*-markers. As in the case of grammatical transformations, the truly productive behavioral transformations are undoubtedly those that combine two or more simpler plans into one. These three chapters make it perfectly plain, however, how difficult it is to formulate a transformational system to achieve the twin goals of empirical adequacy and feasibility of abstract study.

When we ask about the source of our plans, however, we also raise the closely related question of what it might be that stands in the same relation to a plan as a grammar stands to a *P*-marker or as a programming language stands to a particular program. In what form are the rules stored whereby we construct, evaluate, and transform new plans? Probably there are many diverse sets of rules that govern our planning in different enterprises, and only patient observation and analysis of each behavioral system will enable us to describe the rules that govern them.

It is probably no accident that a theory of grammatical structure can be so readily and naturally generalized as a schema for theories of other kinds of complicated human behavior. An organism that is intricate and highly structured enough to perform the operations that we have seen to be involved in linguistic communication does not suddenly lose its intricacy and structure when it turns to nonlinguistic activities. In particular, such an organism can form verbal plans to guide many of its nonverbal acts. The verbal machinery turns out sentences—and, for civilized men, sentences have a compelling power to control both thought and action. Thus the present chapters, even though they have gone well beyond the usual bounds of psychology, raise issues that must be resolved eventually by any satisfactory psychological theory of complicated human behavior.

References

Attneave, F. *Applications of information theory to psychology.* New York: Holt-Dryden, 1959.

Burton, N. G., & Licklider, J. C. R. Long-range constraints in the statistical structure of printed English. *Amer. J. Psychol.*, 1955, **68,** 650–653.

Carnap, R., & Bar-Hillel, Y. *An outline of a theory of semantic information.* Res. Lab. Electronics, Cambridge: Mass. Inst. Tech. Tech. Rept. 247, 1952.

Chapanis, A. The reconstruction of abbreviated printed messages. *J. exp. Psychol.*, 1954, **48,** 496–510.

Cherry, C. *On human communication.* New York: Technology Press and Wiley, 1957.

Chomsky, N. *Logical structure of linguistic theory.* Microfilm. Mass. Inst. Tech. Libraries, 1955.

Condon, E. V. Statistics of vocabulary. *Science,* 1928, **67,** 300.

Cronbach, L. J. On the non-rational application of information measures in psychology. In H. Quastler (Ed.), *Information theory in psychology*. Glencoe, Ill.: Free Press, 1955. Pp. 14–26.

Eifermann, R. R. Negation: a linguistic variable. *Acta Psychol.*, 1961, **18,** 258–273.

Estoup, J. B. *Gammes sténographique*. (4th ed.) Paris: 1916.

Fano, R. M. *The transmission of information*. Res. Lab. Electronics, Cambridge: Mass. Inst. Tech. Tech. Rept. 65, 1949.

Fano, R. M. *The transmission of information*. New York: Wiley, 1961.

Feinstein, A. *Foundations of information theory*. New York: McGraw-Hill, 1958.

Feller, W. *An introduction to probability theory and its applications*. (2nd ed.) New York: Wiley, 1957.

Fletcher, H. *Speech and hearing in communication*. (2nd ed.). New York: Van Nostrand, 1953.

Frick, F. C., & Miller, G. A. A statistical description of operant conditioning. *Amer. J. Psychol.*, 1951, **64,** 20–36.

Frick, F. C., & Sumby, W. H. Control tower language. *J. acoust. Soc. Amer.*, 1952, **24,** 595–597.

Fritz, E. L., & Grier, G. W., Jr. Pragmatic communications: A study of information flow in air traffic control. In H. Quastler (Ed.), *Information theory in psychology*. Glencoe, Ill.: Free Press, 1955. Pp. 232–243.

Garner, W. R. *Uncertainty and structure as psychological concepts*. New York: Wiley, 1962.

Gnedenko, B. V., & Kolmogorov, A. N. *Limit distributions for sums of independent random variables*. Translated by K. L. Chung. Cambridge, Mass.: Addison-Wesley, 1954.

Halle, M., & Stevens, K. N. Analysis by synthesis. In *Proc. Seminar on Speech Compression and Production*, AFCRC-TR-59-198, 1959.

Halle, M., & Stevens, K. N. Speech recognition: A model and a program for research. *IRE Trans. on Inform. Theory*, 1962, **IT-8,** 155–159.

Hardy, G. H., Littlewood, J. E., & Pólya, G. *Inequalities*. (2nd ed.). Cambridge: Cambridge Univer. Press, 1952.

Hartley, R. V. The transmission of information. *Bell System Tech. J.*, 1928, **17,** 535–550.

Hovland, C. I., & Weiss, W. Transmission of information concerning concepts through positive and negative instances. *J. exp. Psychol.*, 1953, **45,** 175–182.

Huffman, D. A. A method for the construction of minimum-redundancy codes. *Proc. IRE*, 1952, **40,** 1098–1101.

Karp, R. M. A note on the application of graph theory to digital computer programming. *Information and Control*, 1960, **3,** 179–190.

Katz, J., & Fodor, J. *The structure of a semantic theory*. To appear in *Language*. Reprinted in J. Katz & J. Fodor. *Readings in the philosophy of language*. New York: Prentice-Hall, 1963.

Khinchin, A. I. *Mathematical foundations of information theory*. Translated by R. A. Silverman and M. D. Friedman. New York: Dover, 1957.

Luce, R. D. *Individual choice behavior*. New York: Wiley, 1959.

Luce, R. D. (Ed.) *Developments in mathematical psychology*. Glencoe, Ill.: Free Press, 1960.

Mandelbrot, B. An informational theory of the structure of language based upon the theory of the statistical matching of messages and coding. In W. Jackson, (Ed.), *Proc. symp. on applications of communication theory*. London: Butterworth, 1953.

Mandelbrot, B. Linguistique statistique macroscopique. In L. Apostel, B. Mandelbrot, & A. Morf. *Logique, langage and théorie de l'information.* Paris: Universitaires de France, 1957. Pp. 1–78.

Mandelbrot, B. Les lois statistique macroscopiques du comportment. *Psychol. Française*, 1958, **3**, 237–249.

Mandelbrot, B. A note on a class of skew distribution functions: Analysis and critique of a paper by H. A. Simon. *Information and Control*, 1959, **2**, 90–99.

Mandelbrot, B. On the theory of word frequencies and on related Markovian models of discourse. In R. Jakobson (Ed.), *Structure of language in its mathematical aspect. Proc. 12th Symp. in App. Math.* Providence, R. I.: American Mathematical Society, 1961. Pp. 190–219.

Markov, A. A. Essai d'une recherche statistique sur le texte du roman "Eugene Onegin," *Bull acad. imper. Sci., St. Petersburg*, 1913, **7.**

Marschak, J. Remarks on the economics of information. In *Contributions to Scientific Research in Management.* Berkeley, Calif.: Univer. of California Press, 1960. Pp. 79–98.

Matthews, G. H. Analysis by synthesis of sentences of natural languages. In *Proc. 1st Int. Cong. on Machine Translation of Languages and Applied Language Analysis, 1961.* Teddington, England: National Physical Laboratory, (in press).

McMillan, B. The basic theorems of information theory. *Ann. math. Stat.*, 1953, **24,** 196–219.

Miller, G. A. *Language and communication.* New York: McGraw-Hill, 1951.

Miller, G. A. What is information measurement? *Amer. Psychologist*, 1953, **8**, 3–11.

Miller, G. A. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychol. Rev.*, 1956, **63**, 81–97.

Miller, G. A. Some effects of intermittent silence. *Amer. J. Psychol.*, 1957, **70**, 311–313.

Miller, G. A. Decision units in the perception of speech. *IRE Trans. Inform. Theory*, 1962, **IT-8**, No. 2, 81–83. (a)

Miller, G. A. Some psychological studies of grammar. *Amer. Psychologist*, 1962 **17**, 748–762. (b)

Miller, G. A., & Frick, F. C. Statistical behavioristics and sequences of responses. *Psychol. Rev.*, 1949, **56**, 311–324.

Miller, G. A., & Friedman, E. A. The reconstruction of mutilated English texts. *Information and Control*, 1957, **1**, 38–55.

Miller, G. A., Galanter, E., & Pribram, K. *Plans and the structure of behavior.* New York: Holt, 1960.

Miller, G. A., Heise, G. A., & Lichten, W. The intelligibility of speech as a function of the context of the test materials. *J. exp. Psychol.*, 1951, **41**, 329–335.

Miller, G. A., & Newman, E. B. Tests of a statistical explanation of the rank-frequency relation for words in written English. *Amer. J. Psychol.*, 1958, **71**, 209–258.

Miller, G. A., Newman, E. B., & Friedman, E. A. Length-frequency statistics for written English. *Information and Control*, 1958, **1**, 370–398.

Miller, G. A., & Selfridge, J. A. Verbal context and the recall of meaningful material. *Amer. J. Psychol.*, 1950, **63**, 176–185.

Newell, A., Shaw, J. C., & Simon, H. A. Report on a general problem-solving program. In *Information Processing. Proc. International Conference on Information Processing, UNESCO*, Paris, June 1959. Pp. 256–264.

Newman, E. B. The pattern of vowels and consonants in various languages. *Amer. J. Psychol.*, 1951, **64**, 369–379.

Pareto, V. *Cours d'economie politique.* Paris: 1897.

Quastler, H. (Ed.). *Information theory in psychology.* Glencoe, Ill.: Free Press, 1955.

Shannon, C. E. A mathematical theory of communication. *Bell System Tech. J.*, 1948, **27**, 379–423.

Shannon, C. E. Prediction and entropy of printed English. *Bell Syst. tech. J.*, 1951, **30**, 50–64.

Skinner, B. F. *Verbal behavior.* New York: Appleton-Century-Crofts, 1957.

Smoke, K. L. Negative instances in concept learning. *J. exp. Psychol.*, 1933, **16**, 583–588.

Somers, H. H. The measurement of grammatical constraints. *Language and Speech*, 1961, **4**, 150–156.

Thorndike, E. L., & Lorge, I. *The teacher's word book of* 30,000 *words.* New York: Bureau of Publications, Teachers College, Columbia University, 1944.

Toda, M. Information-receiving behavior in man. *Psychol. Rev.*, 1956, **63**, 204–212.

Wason, P. C. The processing of positive and negative information. *Quart. J. exp. Psychol.*, 1959, **11**, 92–107.

Wason, P. C. Response to affirmative and negative binary statements. *Brit. J. Psychol.*, 1961, **52**, 133–142.

Wiener, N. *Cybernetics.* New York: Wiley, 1948.

Willis, J. C. *Age and area.* Cambridge: Cambridge Univer. Press, 1922.

Yngve, V. H. A model and an hypothesis for language structure. *Proc. Am. Phil. Soc.*, 1960, **104**, 444–466.

Yngve, V. H. The depth hypothesis. In R. Jakobson (Ed.), *Structure of language and its mathematical aspect. Proc. 12th Symp. in App. Math.* Providence, R. I.: American Mathematical Society, 1961. Pp. 130–138.

Yule, G. U. A mathematical theory of evolution, based on the conclusions of Dr. J. C. Willis, FRS. *Phil. Trans. Roy. Soc. (London)*, 1924, **B 213**, 21–87.

Yule, G. U. *The statistical study of literary vocabulary.* London: Cambridge Univer. Press, 1944.

Ziff, P. *Semantic analysis.* Ithaca: Cornell Univ. Press, 1960.

Zipf, G. K. *The psychobiology of language.* Boston: Houghton-Mifflin, 1935.

Zipf, G. K. *Human behavior and the principle of least effort.* Cambridge, Mass.: Addison-Wesley, 1949.