THE ZERO ERROR CAPACITY OF A NOISY CHANNEL

Claude E. Shannon
Bell Telephone Laboratories, Murray Hill, New Jersey
Massachusetts Institute of Technology, Cambridge, Mass.

Abstract

The zero error capacity C_0 of a noisy channel is defined as the least upper bound of rates at which it is possible to transmit information with zero probability of error. Various properties of C_0 are studied; upper and lower bounds and methods of evaluation of C_0 are given. Inequalities are obtained for the C_0 relating to the "sum" and "product" of two given channels. The analogous problem of zero error capacity C_{0F} for a channel with a feedback link is considered. It is shown that while the ordinary capacity of a memoryless channel with feedback is equal to that of the same channel without feedback, the zero error capacity may be greater. A solution is given to the problem of evaluating C_{0F}.

Introduction

The ordinary capacity C of a noisy channel may be thought of as follows. There exists a sequence of codes for the channel of increasing block length such that the input rate of transmission approaches C and the probability of error in decoding at the receiving point approaches zero. Furthermore, this is not true for any value higher than C. In some situations it may be of interest to consider, rather than codes with probability of error approaching zero, codes for which the probability is zero and to investigate the highest possible rate of transmission (or the least upper bound of these rates) for such codes. This rate, C_0, is the main object of investigation of the present paper. It is interesting that while C_0 would appear to be a simpler property of a channel than C, it is in fact more difficult to calculate and leads to a number of as yet unsolved problems.

We shall consider only finite discrete memoryless channels. Such a channel is specified by a finite transition matrix $\|p_i(j)\|$ where $p_i(j)$ is the probability of input letter i being received as output letter j ($i = 1,2,\ldots,a$; $j = 1,2,\ldots,b$) and $\sum_j p_i(j) = 1$. Equivalently, such a channel may be represented by a line diagram such as Fig. 1.

The channel being memoryless means that successive operations are independent. If the input letters i and j are used, the probability of output letters k and l will be $p_i(k)\,p_j(l)$. A sequence of input letters will be called an input word, a sequence of output letters an output word. A mapping of M messages (which we may take to be the integers $1,2,\ldots,M$) into a subset of input words of length n will be called a block code of length n. $R = \frac{1}{n}\log M$ will be called the input rate for this code. Unless otherwise specified, a code will mean such a block code. We will, throughout, use natural logarithms and natural (rather than binary) units of information, since this simplifies the analytical processes that will be employed.

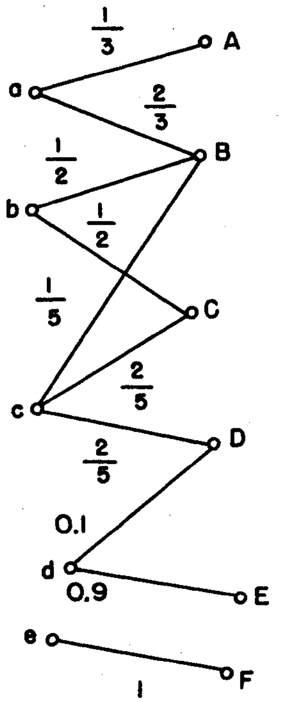

Fig. 1

Why this is called a input rate, why take log here

A decoding system for a block code of length n is a method of associating a unique input message (integer from 1 to M) with each possible output word of length n, that is, a function from output words of length n to the integers 1 to M. The probability of error for a code is the probability when the M input messages are used each with probability 1/M that the noise and the decoding system will lead to an input message different from the one that actually occurred.

If we have two given channels, it is possible to form a single channel from them in two natural ways which we call the sum and product of the two channels. The sum of two

This could kind of based on the decoding algorithm

channels is the channel formed by using inputs from either of the two given channels with the same transition probabilities to the set of output letters consisting of the logical sum of the two output alphabets. Thus the sum channel is defined by a transition matrix formed by placing the matrix of one channel below and to the right of that for the other channel and filling the remaining two rectangles with zeros. If $p_i(j)$ and $\| p_i'(j) \|$ are the individual matrices, the sum has the following matrix:

$$\begin{array}{cccc}
p_1(1) \;.\;.\;. & p_1(r) & 0 \;.\;.\;. & 0 \\
\vdots & \vdots & \vdots & \vdots \\
p_t(1) \;.\;.\;. & p_t(r) & 0 \;.\;.\;. & 0 \\
0 \;.\;.\;. & 0 & p_1'(1) \;.\;.\;. & p_1'(r') \\
\vdots & \vdots & \vdots & \vdots \\
0 \;.\;.\;. & 0 & p_{t'}'(1) \;.\;.\;. & p_{t'}'(r')
\end{array}$$

The product of two channels is the channel whose input alphabet consists of all ordered pairs (i,i') where i is a letter from the first channel alphabet and i' from the second, whose output alphabet is the similar set of ordered pairs of letters from the two individual output alphabets and whose transition probability from (i,i') to (j,j') is $p_i(j)\ p_{i'}'(j')$.

The sum of channels corresponds physically to a situation where either of two channels may be used (but not both), a new choice being made for each transmitted letter. The product channel corresponds to a situation where both channels are used each unit of time. It is interesting to note that multiplication and addition of channels are both associative and commutative, and that the product distributes over a sum. Thus one can develop a kind of algebra for channels in which it is possible to write, for example, a polynomial $\sum a_n K^n$, where the a_n are non-negative integers a and K is a channel. We shall not, however, investigate here the algebraic properties of this system.

The Zero Error Capacity

In a discrete channel we will say that two input letters are adjacent if there is an output letter which can be caused by either of these two. Thus, i and j are adjacent if there exists a t such that both $p_i(t)$ and $p_j(t)$ do not vanish. In Fig. 1, a and c are adjacent, while a and d are not.

If all input letters are adjacent to each other, any code with more than one word has a probability of error at the receiving point greater than zero. In fact, the probability of error in decoding words satisfies

$$P_e \geqslant \frac{M-1}{M}\ p_{\min}^n$$

where $p_{\min}$ is the smallest (non-vanishing) among the $p_i(j)$, n is the length of the code and M is the number of words in the code. To prove this, note that any two words have a possible output word in common, namely the word consisting of the sequence of common output letters when the two input words are compared letter by letter. Each of the two input words has a probability at least $p_{\min}^n$ of producing this common output word. In using the code, the two particular input words will each occur $\frac{1}{M}$ of the time and will cause the common output $\frac{1}{M}\ p_{\min}^n$ of the time. This output can be decoded in only one way. Hence at least one of these situations leads to an error. This error, $\frac{1}{M}\ p_{\min}^n$, is assigned to this code word, and from the remaining $M-1$ code words another pair is chosen. A source of error to the amount $\frac{1}{M}\ p_{\min}^n$ is assigned in similar fashion to one of these, and this is a disjoint event. Continuing in this manner, we obtain a total of at least $\frac{M-1}{M}\ p_{\min}^n$ as probability of error.

If it is not true that the input letters are all adjacent to each other, it is possible to transmit at a positive rate with zero probability of error. The least upper bound of all rates which can be achieved with zero probability of error will be called the zero error capacity of the channel and denoted by C_0. If we let $M_0(n)$ be the largest number of words in a code of length n, no two of which are adjacent, then C_0 is the least upper bound of the numbers $\frac{1}{n} \log M_0(n)$ when n varies through all positive integers.

One might expect that C_0 would be equal to $\log M_0(1)$, that is, that if we choose the largest possible set of non-adjacent letters and form all sequences of these of length n, then this would be the best error free code of length n. This is not, in general, true, although it holds in many cases, particularly when the number of input letters is small. The first failure occurs with five input letters with the channel in Fig. 2. In this channel, it is possible to choose at most two non-adjacent letters, for example 0 and 2. Using sequences of these, 00, 02, 20, and 22 we obtain four words in a code of length two. However, it is possible to construct a code of length two with five members no two of which are adjacent as follows: 00, 12, 24, 31, 43. It is readily verified that no two of these are adjacent. Thus, C_0 for this channel is at least $\frac{1}{2} \log 5$.

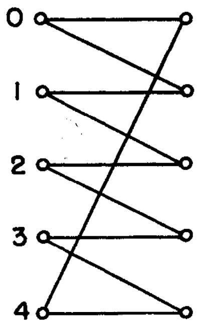

Fig. 2

why does the ordinary capacity kind of depend on
the transition error probability
ordinary capacity is like something with the same adjacent matrix
and any

No method has been found for determining C_0 for the general discrete channel, and this we propose as an interesting unsolved problem in coding theory. We shall develop a number of results which enable one to determine C_0 in many special cases, for example, in all channels with five or less input letters with the single exception of the channel of Fig. 2 (or channels equivalent in adjacency structure to it). We will also develop some general inequalities enabling one to estimate C_0 quite closely in most cases.

It may be seen, in the first place, that the value of C_0 depends only on which input letters are adjacent to each other. Let us define the <u>adjacency</u> <u>matrix</u> for a channel, A_{ij}, as follows.

$$A_{ij} = \begin{cases} 1 \text{ if input letter } i \text{ is adjacent to } j \text{ or} \\ \quad \text{if } i = j \\ 0 \text{ otherwise} \end{cases}$$

Suppose two channels have the same adjacency matrix (possibly after renumbering the input letters of one of them). Then it is obvious that a zero error code for one will be a zero error code for the other and, hence, that the zero error capacity C_0 for one will also apply to the other.

The adjacency structure contained in the adjacency matrix can also be represented as a linear graph. Construct a graph with as many vertices as there are input letters, and connect two distinct vertices with a line or branch of the graph if the corresponding input letters are adjacent. Two examples are shown in Fig. 3, corresponding to the channels of Figs. 1 and 2.

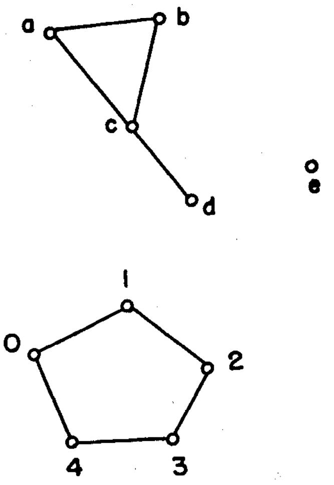

Fig. 3

<u>Theorem 1</u>: The zero error capacity C_0 of a discrete memoryless channel is bounded by the inequalities

$$-\log \min_{P_i} \sum_{ij} A_{ij} P_i P_j \leqslant C_0 \leqslant \min_{p_i(j)} C$$

$$\sum_i P_i = 1, \quad P_i \geqslant 0$$

$$\sum_j p_i(j) = 1, \; p_i(j) \geqslant 0$$

where C is the capacity of any channel with transition probabilities $p_i(j)$ and having the adjacency matrix A_{ij}.

The upper bound is fairly obvious. The zero error capacity is certainly less than or equal to the ordinary capacity for any channel with the same adjacency matrix since the former requires codes with zero probability of error while the latter requires only codes approaching zero probability of error. By minimizing the capacity through variation of the $p_i(j)$ we find the lowest upper bound available through this argument. Since the capacity is a continuous function of the $p_i(j)$ in the closed region defined by $p_i(j) \geqslant 0$, $\sum_j p_i(j) = 1$, we may write min instead of greatest lower bound.

It is worth noting that it is only necessary to consider a particular channel in performing this minimization, although there are an infinite number with the same adjacency matrix. This one particular channel is obtained as follows from the adjacency matrix. If $A_{ik} = 1$ for a pair i k, define an output letter j with $p_i(j)$ and $p_k(j)$ both differing from zero. Now if there are any three input letters, say i k l, all adjacent to each other, define an output letter, say m, with $p_i(m)$ $p_k(m)$ $p_l(m)$ all different from zero. In the adjacency graph this corresponds to a complete sub-graph with three vertices. Next, subsets of four letters or complete subgraphs of four vertices, say i k l m, are given an output letter, each being connected to it, and so on. It is evident that any channel with the same adjacency matrix differs from that just described only by variation in the number of output symbols for some of the pairs, triplets, etc., of adjacent input letters. If a channel has more than one output symbol for an adjacent subset of input letters, then its capacity is reduced by identifying these. If a channel contains no element, say for a triplet i k l of adjacent input letters, this will occur as a special case of our canonical channel which has output letter m for this triplet when $p_i(m)$, $p_k(m)$ and $p_l(m)$ all vanish.

The lower bound of the theorem will now be proved. We use the procedure of random codes based on probabilities for the letters P_i, these being chosen to minimize the quadratic form $\sum_{ij} A_{ij} P_i P_j$. Construct an ensemble of codes each containing M words, each word n letters long. The words in a code are chosen by the following stochastic method. Each letter of each word is chosen independently of all others and is the letter i with probability P_i. We now compute the probability in the ensemble that any particular word is not adjacent to any other word in its code. The probability that the first letter of one word is adjacent to the first letter of a second word is $\sum_{ij} A_{ij}P_iP_j$, since this sums the cases of adjacency with coefficient 1 and those of non-adjacency with coefficient 0. The probability that two words are adjacent in all letters, and therefore adjacent as words, is $(\sum_{ij} A_{ij}P_iP_j)^n$. The probability of non-adjacency is therefore $1 - (\sum_{ij} A_{ij}P_iP_j)^n$. The probability that all M - 1 other words in a code are not adjacent to a given word is, since they are chosen independently,

$$\left[1 - (\sum_{ij} A_{ij}P_iP_j)^n\right]^{M-1}$$

which is, by a well known inequality, greater than $1 - (M - 1)(\sum_{ij} A_{ij}P_iP_j)^n$, which in turn is greater than $1 - M (\sum_{ij} A_{ij}P_iP_j)^n$. If we set $M = (1 - \epsilon)^n(\sum_{ij} A_{ij}P_iP_j)^{-n}$, we then have, by taking ϵ small, a rate as close as desired to $- \log \sum_{ij} A_{ij}P_iP_j$. Furthermore, once ϵ is chosen, by taking n sufficiently large, we can insure that $M(\sum_{ij} A_{ij}P_iP_j)^n = (1 - \epsilon)^n$ is as small as desired, say, less than δ. The probability in the ensemble of codes of a particular word being adjacent to any other in its own code is now less than δ. This implies that there are codes in the ensemble for which the ratio of the number of such undesired words to the total number in the code is less than or equal to δ. For, if not, the ensemble average would be worse than δ. Select such a code and delete from it the words having this property. We have reduced our rate only by at most $\log (1 - \delta)^{-1}$, Since ϵ and δ were both arbitrarily small, we obtain error-free codes arbitrarily close to the rate-log $\min_{P_i} \sum_{ij} A_{ij}P_iP_j$ as stated in the theorem.

In connection with the upper bound of Theorem 1, the following result is useful in evaluating the minimum C. It is also interesting in its own right and will prove useful later in connection with channels having a feedback link.

<u>Theorem 2</u>: In a discrete memoryless channel with transition probabilities $p_i(j)$ and input letter probabilities P_i the following three statements are equivalent.

1) The rate of transmission

$$R = \sum_{i,j} P_i p_i(j) \log (p_i(j)/ \sum_k P_k p_k(j))$$

is stationary under variation of all non-vanishing P_i subject to $\sum_i P_i = 1$ and under variation of $p_i(j)$ for those $p_i(j)$ such that $P_i p_i(j) > 0$ and subject to $\sum_j p_i(j) = 1$.

2) The mutual information between input-output pairs $I_{ij} = \log (p_i(j)/ \sum_k P_k p_k(j))$ is constant, $I_{ij} = I$, for all ij pairs of non-vanishing probability (i.e. pairs for which $P_i p_i(j) > 0$).

3) We have $p_i(j) = r_j$ a function of j only whenever $P_i p_i(j) > 0$; and also $\sum_{i \epsilon S_j} P_i = h$, a constant independent of j where S_j is the set of input letters that can produce output letter j with probability greater than zero. We also have $I = \log h^{-1}$.

The $p_i(j)$ and P_i corresponding to the maximum and minimum capacity when the $p_i(j)$ are varied (keeping, however, any $p_i(j)$ that are zero fixed at zero) satisfy 1), 2) and 3).

<u>Proof</u>: We will show first that 1) and 2) are equivalent and then that 2) and 3) are equivalent.

R is a bounded continouus function of its arguments P_i and $p_i(j)$ in the (bounded) region of allowed values defined by $\sum P_i = 1$, $P_i \geqslant 0$, $\sum_j p_i(j) = 1$, $p_i(j) \geqslant 0$. R has a finite partial derivative with respect to any $p_i(j) > 0$. In fact, we readily calculate

$$\frac{\partial R}{\partial p_i(j)} = P_i \log (p_i(j)/ \sum_k P_k p_k(j))$$

A necessary and sufficient condition that R be stationary for small variation of the non-vanishing $p_i(j)$ subject to the conditions given is that

$$\frac{\partial R}{\partial p_i(j)} = \frac{\partial R}{\partial p_i(k)}$$

for all i, j, k such that P_i, $p_i(j)$, $p_i(k)$ do not vanish. This requires that

$$P_i \log p_i(j) / \sum_m P_m p_m(j) =$$

$$P_i \log p_i(k) / \sum_m P_m p_m(k)$$

If we let $Q_j = \sum_m P_m p_m(j)$, the probability of output letter j, then this is equivalent to

$$\frac{p_i(j)}{Q_j} = \frac{p_i(k)}{Q_k}$$

In other words, $p_i(j)/Q_j$ is independent of j, a function of i only whenever $P_i > 0$ and $p_i(j) > 0$. This function of i we call α_i. Thus

$$p_i(j) = \alpha_i Q_j$$

unless $P_i p_i(j) = 0$.

Now, taking the partial derivative of R with respect to P_i we obtain:

$$\frac{\partial R}{\partial P_i} = \sum_j p_i(j) \log \frac{p_i(j)}{Q_j} - 1$$

For R to be stationary subject to $\sum_i P_i = 1$ we must have $\partial R/\partial P_i = \partial R/\partial P_k$. Thus

$$\sum_j p_i(j) \log \frac{p_i(j)}{Q_j} = \sum_j p_k(j) \log \frac{p_k(j)}{Q_j}$$

Since for $P_i p_i(j) > 0$ we have $p_i(j)/Q_j = \alpha_i$, this becomes

$$\sum_j p_i(j) \log \alpha_i = \sum_j p_k(j) \log \alpha_k$$

$$\log \alpha_i = \log \alpha_k$$

Thus α_i is independent of i and may be written α. Consequently

$$\frac{p_i(j)}{Q_j} = \alpha$$

$$\log \frac{p_i(j)}{Q_j} = \log \alpha = I$$

whenever $P_i p_i(j) > 0$.

The converse result is an easy reversal of the above argument. If

$$\log \frac{p_i(j)}{Q_j} = I, \text{ then}$$

$\partial R/\partial P_i = I - 1$, by a simple substitution in the $\partial R/\partial P_i$ formula. Hence R is stationary under variation of P_i constrained by $\sum P_i = 1$. Further, $\partial R/\partial p_i(j) = P_i I = \partial R/\partial p_i(k)$, and hence the variation of R also vanishes subject to $\sum_j p_i(j) = 1$.

We now prove that 2) implies 3). Suppose $\log \frac{p_i(j)}{Q_j} = I$ whenever $P_i p_i(j) > 0$. Then $p_i(j) = e^I Q_j$, a function of j only under this same condition. Also, if $q_j(i)$ is the conditional probability of i given j, then

$$\frac{Q_j q_j(i)}{P_i Q_j} = e^I$$

$$q_j(i) = e^I P_i$$

$$1 = \sum_{i \epsilon S_j} q_j(i) = e^I \sum_{i \epsilon S_j} P_i$$

To prove that 3) implies 2) we assume

$$p_i(j) = r_j$$

when $P_i p_i(j) > 0$. Then

$$\frac{P_i p_i(j)}{P_i Q_j} = \frac{r_j}{Q_j} = \lambda_j \text{ (say)} = \frac{Q_j q_j(i)}{P_i Q_j} = \frac{q_j(i)}{P_i}$$

Now, summing the equation $P_i \lambda_j = q_j(i)$ over $i \epsilon S_j$ and using the assumption from 3) that $\sum_{S_j} P_i = h$ we obtain

$$h \lambda_j = 1$$

so λ_j is h^{-1} and independent of j. Hence $I_{ij} = I = \log h^{-1}$.

The last statement of the theorem concerning minimum and maximum capacity under variation of $p_i(j)$ follows from the fact that R at these points must be stationary under variation of all non-vanishing P_i and $p_i(j)$, and hence the corresponding P_i and $p_i(j)$ satisfy condition 1) of the theorem.

For simple channels it is usually more convenient to apply particular tricks in trying to evaluate C_0 instead of the bounds given in Theorem 1, which involve maximizing and minimizing processes. The simplest lower bound, as mentioned before, is obtained by merely finding the logarithm of the maximum number of non-adjacent input letters.

A very useful device for determining C_0 which works in many cases may be described using the notion of an adjacency-reducing mapping.

By this we mean a mapping of letters into other letters, $i \rightarrow \alpha(i)$, with the property that if i and j are not adjacent in the channel (or graph) then $\alpha(i)$ and $\alpha(j)$ are not adjacent. If we have a zero-error code, then we may apply such a mapping letter by letter to the code and obtain a new code which will also be of the zero-error type, since no adjacencies can be produced by the mapping.

Theorem 3: If all the input letters i can be mapped by an adjacency-reducing mapping $i \rightarrow \alpha(i)$ into a subset of the letters no two of which are adjacent, then the zero-error capacity C_o of the channel is equal to the logarithm of the number of letters in this subset.

For, in the first place, by forming all sequences of these letters we obtain a zero-error code at this rate. Secondly, any zero error code for the channel can be mapped into a code using only these letters and containing, therefore, at most $e^{C_o n}$ non-adjacent words.

The zero-error capacities, or, more exactly, the equivalent numbers of input letters for all adjacency graphs up to five vertices are shown in Fig. 4. These can all be found readily by the method of Theorem 3, except for the channel of Fig. 2 mentioned previously, for which we know only that the zero-error capacity lies in the range $\frac{1}{2} \log 5 \leq C_o \leq \log \frac{5}{2}$.

All graphs with six vertices have been examined and the capacities of all of these can also be found by this theorem, with the exception of four. These four can be given in terms of the capacity of Fig. 2, so that this case is essentially the only unsolved problem up to seven vertices. Graphs with seven vertices have not been completely examined but at least one new situation arises, the analog of Fig. 2 with seven input letters.

As examples of how the N_o values were computed by the method of adjacency-reducing mappings, several of the graphs in Fig. 4 have been labelled to show a suitable mapping. The scheme is as follows. All nodes labelled a are mapped into node α as well as α itself. All nodes labelled b and also β are mapped into node β. All nodes labelled c and γ are mapped into node γ. It is readily verified that no new adjacencies are produced by the mappings indicated and that the α, β, γ nodes are non-adjacent.

C_o for Sum and Product Channels

Theorem 4: If two memoryless channels have zero-error capacities $C_o' = \log A$ and $C_o'' = \log B$, their sum has a zero-error capacity greater than or equal to log (A + B) and their product a zero error capacity greater than or equal to $C_o' + C_o''$. If the graph of either of the two channels can be reduced to non-adjacent points by the mapping method (Theorem 3), then these inequalities can be replaced by equalities.

Proof: It is clear that in the case of the product, the zero error capacity is at least $C_o' + C_o''$, since we may form a product code from two codes with rates close to C_o' and C_o''. If these codes are not of the same length, we use for the new code the least common multiple of the individual lengths and form all sequences of the code words of each of the codes up to this length. To prove equality in case one of the graphs, say that for the first channel, can be mapped into A non-adjacent points, suppose we have a code for the product channel. The letters for the product code, of course, are ordered pairs of letters corresponding to the original channels. Replace the first letter in each pair in all code words by the letter corresponding to reduction by the mapping method. This reduces or preserves adjacency between words in the code. Now sort the code words into A^n subsets according to the sequences of first letters in the ordered pairs. Each of these subsets can contain at most B^n members, since this is the largest possible number of codes for the second channel of this length. Thus, in total, there are at most $A^n B^n$ words in the code, giving the desired result.

In the case of the sum of the two channels, we first show how, from two given codes for the two channels, to construct a code for the sum channel with equivalent number of letters equal to $A^{1-\delta} + B^{1-\delta}$, where δ is arbitrarily small and A and B are the equivalent number of letters for the two codes. Let the two codes have lengths n_1 and n_2. The new code will have length n where n is the smallest integer greater than both $\frac{n_1}{\delta}$ and $\frac{n_2}{\delta}$. Now form codes for the first channel and for the second channel for all lengths k from zero to n as follows. Let k equal $an_1 + b$, where a and b are integers and $b < n_1$. We form all sequences of a words from the given code for the first channel and fill in the remaining b letters arbitrarily, say all with the first letter in the code alphabet. We achieve at least $A^{k - \delta n}$ different words of length k none of which is adjacent to any other. In the same way we form codes for the second channel and achieve $B^{k - \delta n}$ words in this code of length k. We now intermingle the k code for the first channel with the n - k code for the second channel in all $\binom{n}{k}$ possible ways and do this for each value of k. This produces a code n letters long with at least $\sum_{k=o}^{n} \binom{n}{k} A^{k - n\delta} B^{n-k-n\delta}$

$= (AB)^{-\delta n}(A + B)^n$ different words. It is readily seen that no two of these different words are adjacent. The rate is at least $\log (A + B) - \delta \log AB$, and since δ was arbitrarily small, we can achieve a rate arbitrarily close to log (A + B).

To show that it is not possible, when one of the graphs reduces by mapping to non-adjacent points, to exceed the rate corresponding to the number of letters A + B, consider any given code of length n for the sum channel. The words in this consist of sequences of letters each letter corresponding to one or the other of the two

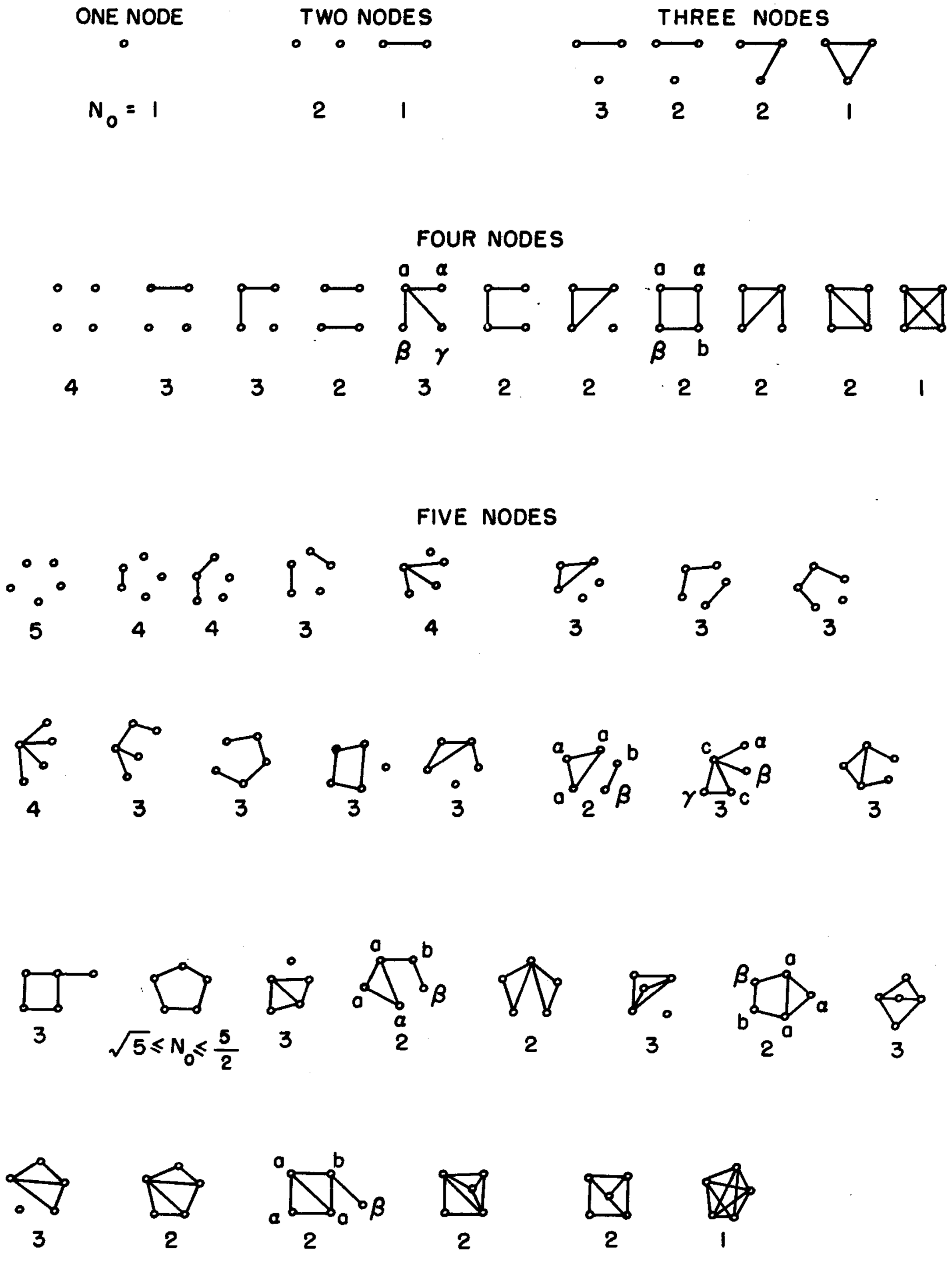

Fig. 4 - All graphs with 1, 2, 3, 4, 5 nodes and the corresponding N_0 for channels with these as adjacency graphs (note $C_0 = \log N_0$)

channels. The words may be subdivided into classes corresponding to the pattern of the choices of letters between the two channels. There are 2^n such classes with $\binom{n}{k}$ classes in which exactly k of the letters are from the first channel and n - k from the second. Consider now a particular class of words of this type. Replace the letters from the first channel alphabet by the corresponding non-adjacent letters. This does not harm the adjacency relations between words in the code. Now, as in the product case, partition the code words according to the sequence of letters involved from the first channel. This produces at most A^k subsets. Each of these subsets contains at most $B^{n - k}$ members, since this is the greatest possible number of non-adjacent words for the second channel of length n - k. In total, then, summing over all values of k and taking account of the $\binom{n}{k}$ classes for each k, there are at most $\sum_k \binom{n}{k} A^k B^{n-k}$ $=(A + B)^n$ words in the code for the sum channel. This proves the desired result.

Theorem 4, of course, is analogous to known results for ordinary capacity C, where the product channel has the sum of the ordinary capacities and the sum channel has an equivalent number of letters equal to the sum of the equivalent numbers of letters for the individual channels. We conjecture but have not been able to prove that the equalities in Theorem 4 hold in general, not just under the conditions given. We now prove a lower bound for the probability of error when transmitting at a rate greater than C_0.

<u>Theorem 5</u>: In any code of length n and rate $R > C_0$, $C_0 > 0$, the probability of error P_e will satisfy $P_e \geq (1 - e^{-n(C_0 - R)})\, p^n_{min}$, where p_{min} is the minimum non-vanishing $p_i(j)$.

<u>Proof</u>: By definition of C_0 there are not more than e^{nC_0} non-adjacent words of length n. With $R > C_0$, among e^{nR} words there must, therefore, be an adjacent pair. The adjacent pair has a common output word which either can cause with a probability at least p^n_{min}. This output word cannot be decoded into both inputs. At least one, therefore, must cause an error when it leads to this output word. This gives a contribution at least $e^{-nR}\, p^n_{min}$ to the probability of error P_e. Now omit this word from consideration and apply the same argument to the remaining $e^{nR} - 1$ words of the code. This will give another adjacent pair and another contribution of error of at least $e^{-nR}\, p^n_{min}$. The process may be continued until the number of code points remaining is just e^{nC_0}. At this time, the computed probability of error must be at least $(e^{nR} - e^{nC_0})e^{-nR}\, p^n_{min}$

$= (1 - e^{n(C_0 - R)})\, p^n_{min}$.

<u>Channels with a Feedback Link</u>

We now consider the corresponding problem for channels with complete feedback. By this we mean that there exists a return channel sending back from the receiving point to the transmitting point, without error, the letters actually received. It is assumed that this information is received at the transmitting point before the next letter is transmitted, and can be used, therefore, if desired, in choosing the next transmitted letter.

It is interesting that for a memoryless channel the ordinary forward capacity is the same with or without feedback. This will be shown in Theorem 6. On the other hand, the zero error capacity may, in some cases, be greater with feedback than without. In the channel shown in Fig. 5, for example, $C_0 = \log 2$. However, we will see as a result of Theorem 7 that with feedback the zero error capacity $C_{0F} = \log 2.5$.

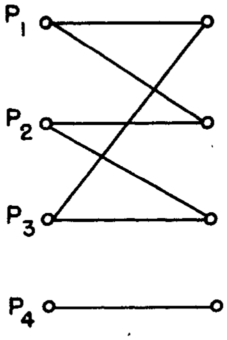

Fig. 5

We first define a block code of length n for a feedback system. This means that at the transmitting point there is a device with two inputs, or, mathematically, a function with two arguments. One argument is the message to be transmitted, the other, the past received letters (which have come in over the feedback link). The value of the function is the next letter to be transmitted. Thus, the function may be thought of as $x_{j+1} = f(k, v_j)$ where x_{j+1} is the j + 1 transmitted letter in a block, k is an index ranging from 1 to M, and represents the specific message, and v_j is a received word of length j. Thus j ranges from 0 to n - 1 and v_j over all received words of these lengths.

In operation, if message m_k is to be sent f is evaluated for f(k -) where the - means "no word" and this is sent as the first transmitted letter. If the feedback link sends back α, say, as the first received letter, the next transmitted letter will be $f(k, \alpha)$. If this is received as β. the next transmitted letter will be $f(k, \alpha\beta)$, etc.

<u>Theorem 6</u>: In a memoryless discrete channel with feedback, the forward capacity is equal to the ordinary capacity C (without feedback). The average change in mutual information I_{vm} between received sequence v and message m for a letter of text is not greater than C.

<u>Proof</u>: Let v be the received sequence to date of a block, m the message, x the next transmitted letter and y the next received letter. These are all random variables and, also, x is a function of m and v. This function, namely, is the one which defines the encoding procedure with feedback whereby the next transmitted letter x is determined by the message m and the feedback information v from the previous received signals. The channel being memoryless implies that the next operation is independent of the past, in particular, $\Pr[y/x] = \Pr[y/x,v]$.

The average change in mutual information, when a particular v has been received, due to the x,y pair is given by (we are averaging over messages m and next received letters y, for a given v):

$$\overline{\Delta I} = \overline{I_{m,vy}} - \overline{I_{m,v}} = \sum_{y,m} \Pr[y,m/v] \cdot \log \frac{\Pr[v,y,m]}{\Pr[v,y]\Pr[m]} - \sum_{m} \Pr[m/v] \cdot \log \frac{\Pr[v,m]}{\Pr[v]\Pr[m]}$$

Since $\Pr[m/v] = \sum_{y} \Pr[y,m/v]$, the second sum may be rewritten as $\sum_{y,m} \Pr[y,m/v] \log \frac{\Pr[v,m]}{\Pr[v]\,\Pr[m]}$

The two sums then combine to give

$$\overline{\Delta I} = \sum_{y,m} \Pr[y,m/v] \log \frac{\Pr[v,y,m]\,\Pr[v]}{\Pr[v,m]\,\Pr[v,y]}$$

$$= \sum_{y,m} \Pr[y,m/v] \log \frac{\Pr[y/v,m]\,\Pr[v]}{\Pr[v,y]}$$

The sum on m may be thought of as summed first on the m's which result in the same x (for the given v), recalling that x is a function of m and v, and then summing on the different x's. In the first summation, the term $\Pr[y/v,m]$ is constant at $\Pr[y/x]$ and the coefficient of the logarithm sums to $\Pr[x,y/v]$. Thus we can write

$$\Delta I = \sum_{x,y} \Pr[x,y/v] \log \frac{\Pr[y/x]}{\Pr[y/v]}$$

Now consider the rate for the channel (in the ordinary sense without feedback) if we should assign to the x's the probabilities $q(x) = \Pr[x/v]$. The probabilities for pairs, $r(x,y)$, and for the y's alone, $w(y)$, in this situation would then be

$$\begin{aligned} r(x,y) &= q(x) \Pr[y/x] \\ &= \Pr[x/v] \Pr[y/x] \\ &= \Pr[x,y/v] \\ w(y) &= \sum_{x} r(x,y) \\ &= \sum_{x} \Pr[x,y/v] \\ &= \Pr[y/v] \end{aligned}$$

Hence the rate would be

$$\begin{aligned} R &= \sum_{x,y} r(x,y) \log \frac{\Pr[y/x]}{w(y)} \\ &= \sum_{x,y} \Pr[x,y/v] \log \frac{\Pr[y/x]}{\Pr[y/v]} \\ &= \Delta I \end{aligned}$$

Since $R \leq C$, the channel capacity (C being the maximum possible R for all q(x) assignments), we conclude that

$$\Delta I \leq C.$$

Since the average change in I per letter is not greater than C, the average change in n letters is not greater than nC. Hence, in a block code of length n with input rate R, if $R > C$ then the equivocation at the end of a block will be at least R - C, just as in the non-feedback case. In other words, it is not possible to approach zero equivocation (or, as easily follows, zero probability of error) at a rate exceeding the channel capacity. It is, of course, possible to do this at rates less than C, since certainly anything that can be done without feedback can be done with feedback.

It is interesting that the first sentence of Theorem 6 can be generalized readily to channels with memory provided they are of such a nature that the internal state of the channel can be calculated at the transmitting point from the initial state and the sequence of letters that have been transmitted. If this is not the case, the conclusion of the theorem will not always be true, that is, there exist channels of a more complex sort for which the forward capacity with feedback exceeds that without feedback. We shall not, however, give the details of these generalizations here.

Returning now to the zero-error problem, we define a zero error capacity C_{oF} for a channel with feedback in the obvious way--the least upper bound of rates for block codes with no errors. The next theorem solves the problem of evaluating C_{oF} for memoryless channels with feedback, and indicates how rapidly C_{oF} may be approached as the block length n increases.

<u>Theorem 7</u>: In a memoryless discrete channel with complete feedback of received letters to the transmitting point, the zero error capacity C_{oF} is zero if all pairs of input letters are adjacent. Otherwise $C_{oF} = \log P_o^{-1}$ where

$$P_o = \min_{P_i} \max_{j} \sum_{i \epsilon S_j} P_i$$

P_i being a probability assigned to input letter i ($\sum_i P_i = 1$) and S_j the set of input letters which can cause output letter j with probability greater than zero. A zero error block code of length n can be found for such a feedback channel which transmits at a rate $R \geq C_{oF} (1 - \frac{2}{n} \log_2 2t)$ where t is the number of input letters.

The P_o occuring in this theorem has the following meaning. For any given assignment of probabilities P_i to the input letters one may calculate, for each output letter j, the total probability of all input letters that can (with positive probability) cause j. This is $\sum_{i \epsilon S_j} P_i$. Output letters for which this is large may be thought of as "bad" in that when received there is a large uncertainty as to the cause. To obtain P_o one adjusts the P_i so that worst output letter in this sense is as good as possible.

We first show that if all letters are adjacent to each other $C_{oF} = 0$. In fact, in any coding system, any two messages, say m_1 and m_2 can lead to the same received sequence with positive probability. Namely, the first transmitted letters corresponding to m_1 and m_2 have a possible received letter in common. Assuming this occurs, calculate the next transmitted letters in the coding system for m_1 and m_2. These also have a possible received letter in common. Continuing in this manner we establish a received word which could be produced by either m_1 or m_2 and therefore they cannot be distinguished with certainty.

Now consider the case where not all pairs are adjacent. We will first prove, by induction on the block length n, that the rate $\log P_o^{-1}$ cannot be exceeded with a zero error code. For n = o the result is certainly true. The inductive hypothesis will be that no block code of length n - 1 transmits at a rate greater than $\log P_o^{-1}$, or, in other words, can resolve with certainty more than

$$e^{(n-1) \log P_o^{-1}} = P_o^{-(n-1)}$$

different messages. Now suppose (in contradiction to the desired result) we have a block code of length n resolving M messages with $M > P_o^{-n}$. The first transmitted letter for the code partitions these M messages among the input letters for the channel. Let F_i be the fraction of the messages assigned to letter i (that is, for which i is the first transmitted letter). Now these F_i are like probability assignments to the different letters and therefore by definition of P_o, there is some output letter, say letter k, such that $\sum_{i \epsilon S_k} F_i \geq P_o$. Consider the set of messages for which the first transmitted letter belongs to S_k. The number of messages in this set is at least $P_o M$. Any of these can cause output letter k as first received letter. When this happens there are n - 1 letters yet to be transmitted and since $M > P_o^{-n}$ we have $P_o M > P_o^{-(n-1)}$. Thus we have a zero error code of block length n - 1 transmitting at a rate greater than $\log P_i^{-1}$, contradicting the inductive assumption. Note that the coding function for this code of length n - 1 is formally defined from the original coding function by fixing the first received letter at k.

We must now show that the rate $\log P_o^{-1}$ can actually be approached as closely as desired with zero error codes. Let P_i be the set of probabilities which, when assigned to the input letters, give P_o for $\min_{P_i} \max_{j} \sum_{i \epsilon S_j} P_i$. The general scheme of the code will be to divide the M original messages into t different groups corresponding to the first transmitted letter. The number of messages in these groups will be approximately proportional to $P_1, P_2, \ldots P_t$. The first transmitted letter, then, will correspond to the group containing the message to be transmitted. Whatever letter is received, the number of possible messages compatible with this received letter will be approximately P_oM. This subset of possible messages is known both at the receiver and (after the received letter is sent back to the transmitter) at the transmitting point.

The code system next subdivides this subset of messages into t groups, again approximately in proportion to the probabilities P_i. The second letter transmitted is that corresponding to the group containing the actual message. Whatever letter is received, the number of messages compatible with the two received letters is now, roughly, P_o^2M.

This process is continued until only a few messages (less than t^2) are compatible with all the received letters. The ambiguity among these is then resolved by using a pair of non-adjacent letters in a simple binary code. The code thus constructed will be a zero error code for the channel.

Our first concern is to estimate carefully the approximation involved in subdividing the messages into the t groups. We will show that for any M and any set of P_i $\sum P_i = 1$, it is possible to subdivide the M messages into groups of $m_1, m_2, \ldots m_t$ such that $m_i = o$ whenever $P_i = o$ and

$$\left| \frac{m_i}{M} - P_i \right| \leqslant \frac{1}{M} \qquad i = 1, \ldots, t$$

We assume without loss of generality that $P_1, P_2, \ldots P_s$ are the non-vanishing P_i. Choose m_1 to be the largest integer such that $\frac{m_1}{M} \leqslant P_1$. Let $P_1 - \frac{m_1}{M} = \delta_1$. Clearly $|\delta_1| \leqslant \frac{1}{M}$. Next choose m_2 to be the smallest integer such that $\frac{m_2}{M} \geqslant P_2$ and let $P_2 - \frac{m_2}{M} = \delta_2$. We have $|\delta_2| \leqslant \frac{1}{M}$. Also $|\delta_1 + \delta_2| \leqslant \frac{1}{M}$, since δ_1 and δ_2 are opposite in sign and each less than $\frac{1}{M}$ in absolute value. Next, m_3 is chosen so that $\frac{m_3}{M}$ approximates, to within $\frac{1}{M}$, to P_3. If $\delta_1 + \delta_2 \geqslant 0$, then $\frac{m_3}{M}$ is chosen less than or equal to P_3. If $\delta_1 = \delta_2 < 0$, then $\frac{m_3}{M}$ is chosen greater than or equal to P_3. Thus again $P_3 - \frac{m_3}{M} = \delta_3 \leqslant \frac{1}{M}$ and $|\delta_1 + \delta_2 + \delta_3| \leqslant \frac{1}{M}$. Continuing in this manner through P_{s-1} we obtain approximationsfor $P_1, P_2, \ldots, P_{s-1}$ with the property that $|\delta_1 + \delta_2 + \ldots + \delta_{s-1}| \leqslant \frac{1}{M}$, or

$$|M(P_1 + P_2 + \ldots + P_{s-1}) - (m_1 + m_2 + \ldots + m_{s-1})| \leqslant 1.$$

If we now define m_s as $M - \sum_1^{s-1} m_i$ then this inequality can be written $|M(1 - P_s) - (M - m_s)| \leqslant 1$. Hence $\left| \frac{m_s}{M} - P_s \right| \leqslant \frac{1}{M}$. Thus we have achieved the objective of keeping all approximation $\frac{m_i}{M}$ to within $\frac{1}{M}$ of P_i and having $\sum m_i = M$.

Returning now to our main problem note first that if $P_o = 1$ then $C_{oF} = 0$ and the theorem is trivially true. We assume, then, that $P_o < 1$. We wish to show that $P_o \leqslant (1 - \frac{1}{t})$. Consider the set of input letters which have the maximum value of P_i. This maximum is certainly greater than or equal to the average $\frac{1}{t}$. Furthermore, we can arrange to have at least one of these input letters not connected to some output letter. For suppose this is not the case. Then either there are no other input letters beside this set and we contradict the assumption that $P_o < 1$, or there are other input letters with smaller values of P_i. In this case, by reducing the P_i for one input letter in the maximum set and increasing correspondingly that for some input letter which does not connect to all output letters, we do not increase the value of P_o (for any S_j) and create an input letter of the desired type. By consideration of an output letter to which this input letter does not connect we see that $P_o \leqslant 1 - \frac{1}{t}$.

Now suppose we start with M messages and subdivide into groups approximating proportionality to the P_i as described above. Then when a letter has been received, the set of possible messages (compatible with this received letter) will be reduced to those in the groups corresponding to letters which connect to the actual received letter. Each output letter connects to not more than t - 1 input letters (otherwise we would have $P_o = 1$). For each of the connecting groups, the error in approximating P_i has been less than or equal to $\frac{1}{M}$. Hence the total relative number in all connecting groups for any output letter is less than or equal to $P_o + \frac{t-1}{M}$. The total number of possible messages after receiving the first letter consequently drops from M to a number less than or equal to $P_oM + t - 1$.

In the coding system to be used, this remaining possible subset of messages is subdivided again among the input letters to approximate in the same fashion the probabilities P_i. This subdivision can be carried out both at receiving point and transmitting point using the same standard procedure (say, exactly the one described above) since with the feedback both terminals have available the required data, namely the first received letter.

The second transmitted letter obtained by this procedure will again reduce at the receiving point the number of possible messages to a value not greater than P_0 $(P_0M + t - 1) + t - 1$. This same process continues with each transmitted letter. If the upper bound on the number of possible remaining messages after k letters is M_k, then $M_{k+1} = P_o M_k + t - 1$. The solution of this difference equation is

$$M_k = AP_o^k + \frac{t - 1}{1 - P_o}$$

This may be readily verified by substitution in the difference equation. To satisfy the initial conditions $M_o = M$ requires $A = M - \frac{t-1}{1 - P_o}$. Thus the solution becomes

$$M_k = (M - \frac{t-1}{1 - P_o}) P_o^k + \frac{t-1}{1 - P_o}$$

$$= M P_o{}^k + \frac{t-1}{1 - P_o}(1 - P_o{}^k)$$

$$\leq M P_o{}^k + t (t - 1)$$

since we have seen above that $1 - P_o \geq \frac{1}{t}$.

If the process described is carried out for n_1 steps, where n_1 is the smallest integer $\geq$ d where d is the solution of $MP_o{}^d = 1$, then the number of possible messages left consistent with the received sequence will be not greater than $1 + t (t - 1) \leq t^2$ (since $t > 1$, otherwise we should have $C_{oF} = 0$). Now the pair of non-adjacent letters assumed in the theorem may be used to resolve the ambiguity among these t^2 or less messages. This will require not more than $1 + \log_2 t^2 = \log_2 2t^2$ additional letters. Thus, in total, we have used not more than $d + 1 + \log_2 2t^2 = d + \log_2 4t^2 = n$ say as block length. We have transmitted in this block length a choice from $M = P_o^{-d}$ messages. Thus the zero error rate we have achieved is

$$R = \frac{1}{n} \log M \geq \frac{d \log P_o^{-1}}{d + \log_2 4t^2}$$

$$= (1 - \frac{1}{n} \log 4t^2) \log P_o^{-1}$$

$$= (1 - \frac{1}{n} \log 4t^2) \quad C_{oF}$$

Thus we can approximate to C_{oF} as closely as desired with zero error codes.

As an example of Theorem 7 consider the channel in Fig. 5. We wish to evaluate P_o. It is easily seen that we may take $P_1 = P_2 = P_3$ in forming the min max of Theorem 7, for if they are unequal the maximum $\sum_{i \epsilon S_j} P_i$ for the corresponding three output letters would be reduced by equalizing. Also it is evident, then, that $P_4 = P_1 + P_2$, since otherwise a shift of probability one way or the other would reduce the maximum. We conclude, then, that $P_1 = P_2 = P_3 = 1/5$ and $P_4 = 2/5$. Finally, the zero error capacity with feedback is $\log P_o^{-1} = \log 5/2$.

There is a close connection between the min max process of Theorem 7 and the process of finding the minimum capacity for the channel under variation of the non-vanishing transition probabilities $p_i(j)$ as in Theorem 2. It was noted there that at the minimum capacity each output letter can be caused by the same total probability of input letters. Indeed, it seems very likely that the probabilities of input letters to attain the minimum capacity are exactly those which solve the min max problem of Theorem 7, and, if this is so, the $C_{min} = \log P_o^{-1}$.

Acknowledgement

I am indebted to Peter Elias for first pointing out that a feedback link could increase the zero-error capacity, as well as for several suggestions that were helpful in the proof of Theorem 7.