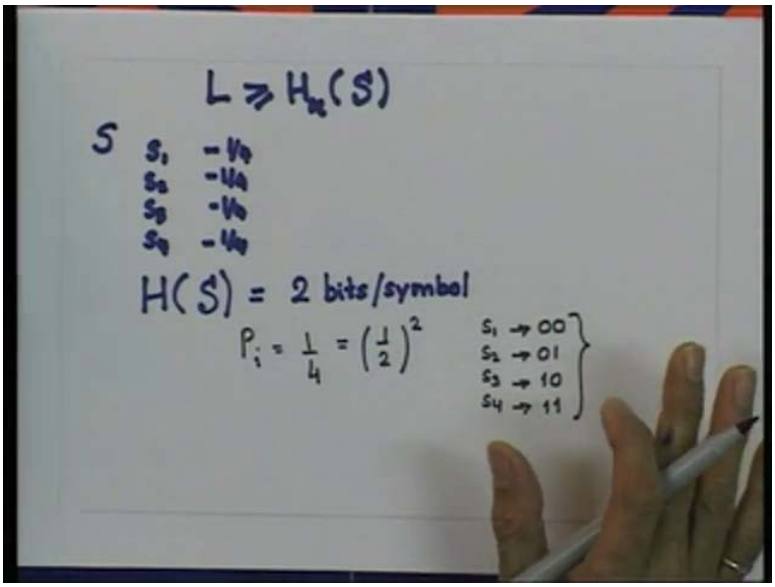
Information Theory And Coding
Prof. S. N. Merchant
Electrical Engineering
Indian institute of Technology, Bombay

Lecture – 10
Shannon's First Theorem

In the last class, we derived a very important result in information theory, which states that the average length of a code can never be greater than the entropy of a source.

(Refer Slide Time: 01:01)

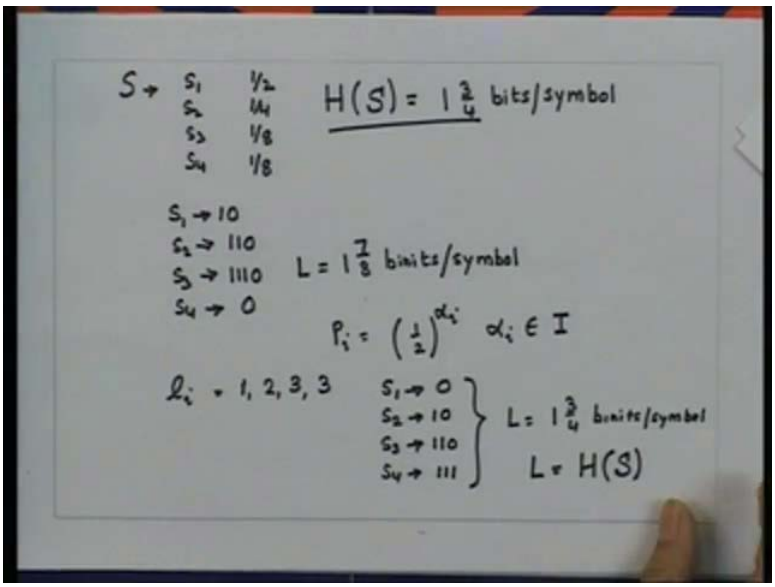


So, what we derived was average length has to be always greater than equal to the entropy of the source measured in array units. Now, to appreciate the importance of this result let us re visit some of the examples which we had studied in earlier in our course. So, one example which we which we which we had considered was that we had a source s consisting of 4 symbols s1, s2, s3, s4 each of this symbols are equiprobable. So, we know that the entropy of the source measured with the base 2 is equal to 2 bits per symbol.

Now, if I try to design any code a binary instantaneous code than my length of that code can never be less than 2 bits per symbol. And in this case each of this probabilities P i which is equal to 1 by 4 is of the form half raised to two. So, what it implies that I can design a compact code with 4 code words each of length 2 and 1 such code is. So, now the length of this code is also 2 binits per symbol. So, there exists no uniquely decodable

code for this source with smaller average code word length, that is the length of 2 binits per symbol. Now, let us look at one more example to understand the derivation of that important result that is L is greater than or equal to entropy of a source.

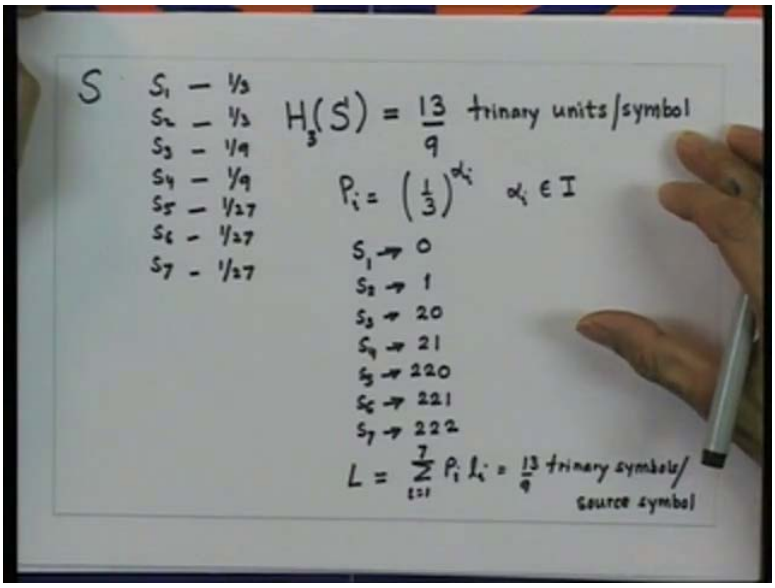
(Refer Slide Time: 03:58)



So, if I take this same source s with the same 4 symbols, but with different probabilities given as half one forth, one eighth, one eight h then the entropy for this source will turn out to be 1 3 4 bits per symbol. Earlier in our course we had designed a instantaneous code for the source and that code was given as S1 is 1 0, S2 is 1 1 0, S is 1 1 1 0 and S 4 is 0. This code is uniquely decodable code in fact it is a instantaneous code. Now, to calculate the length for this code it will turn out to be equal to 1 of 7 by 8 binits per symbol.

So, even in this case we find the length of the code greater than the entropy of the source, but now each P i in this case also is of the form P i is of the form half raised to alpha i where alpha i belongs to integer. Therefore, it is possible to achieve the lower bound of 1 3 4 binits per symbol and this is done by setting Li equal to 1 2 3 and 3 respectively for this 4 symbol S1, S2, S3 and S4. So, the code would be S1 0, S2 1 0, S3 1 1 0 and S4 1 1 1, if the length the average length for this code will come out to be 1 3 4 binits per symbol. So, again in this case we find that the length average length turns out to be equal to the entropy of the source. And as a final example to explain the importance of Lis equal to H s let us consider a source s with seven symbols.

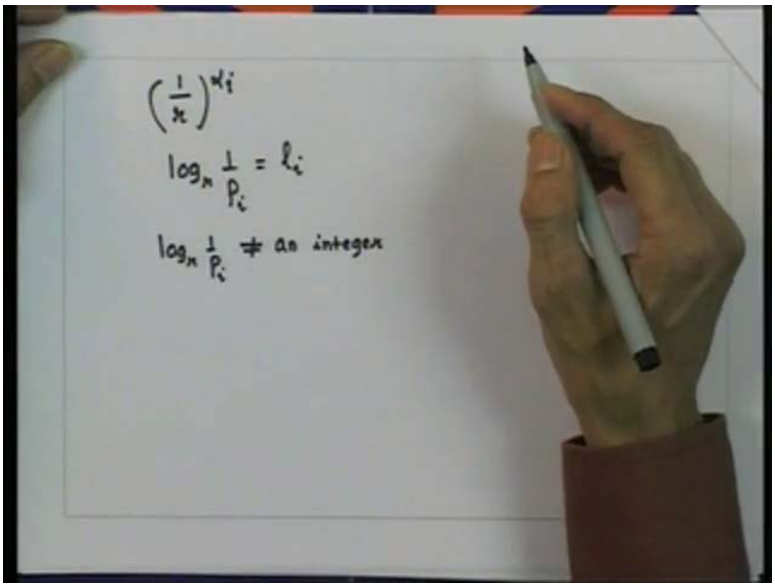
(Refer Slide Time: 07:13)



So I have a source s consisting of seven symbols S1, S2, S3, S4, S5, S 6, S7 each with symbol probabilities given as one third, 1 third, 1 by 9, 1 by 9, 1 by 9, 1 by 7. Now, the entropy for this source will take in the 3 array units we calculate that will turn out to be 13 by 9 trinary units per symbol. And again we find that the symbol probabilities are of the form one third alpha i where alpha i belongs to integer.

And we can once we have the lengths alpha i are nothing but the lengths for the code and we can design the instantaneous code as follows S1 is 0 S2 1. Now, if you calculate the length for this code as P i Li Lis equal to 1 to 7 will turn out to be 13 by 9 trinary symbols per source symbol. So, far what we have seen is that we have looked at the coding problem for the 0 memory source with symbol probabilities of the form 1 by r alpha i.

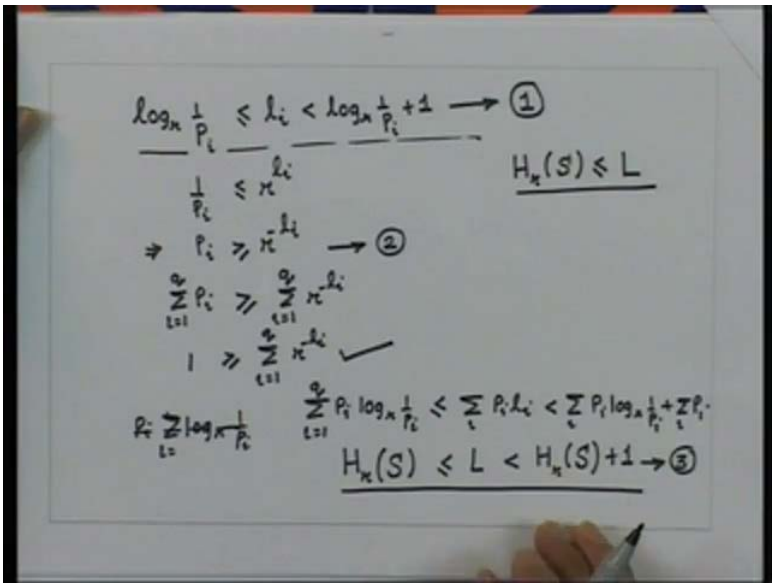
(Refer Slide Time: 09:56)



So, if I have log r 1 by P i is equal to Li. If I have my P i of this form then I can write log r 1 P i is equal to Li where I choose my Li equal to alpha i. Now, the next question arises is that if this condition is not satisfied then how do I choose my length. Now, what it means that if log of r 1 by P i is not an integer than how do I choose my length.

It might seem reasonable that a compact code could be formed by choosing Li as the first integer which is greater than log r 1 by P i. Now, this tempting conjecture is not valid if we select Li in this manner it is not necessary that we will get a compact code, but selecting Li in this manner where Li is the integer, which is just larger than log r 1 by P i can lead to some important results. So, let us select Li, therefore as the unique integer satisfying this condition.

(Refer Slide Time: 12:01)



So, what we will do is we will select Li as a integer which is just larger than this value it means that it follows this inequality. So, if I choose a set of Li which follow this inequality then the next question is it possible for me to design or synthesize an instantaneous code which uses this set of Li. So, that question can be answered that question can be answered if I can test krafts inequality.

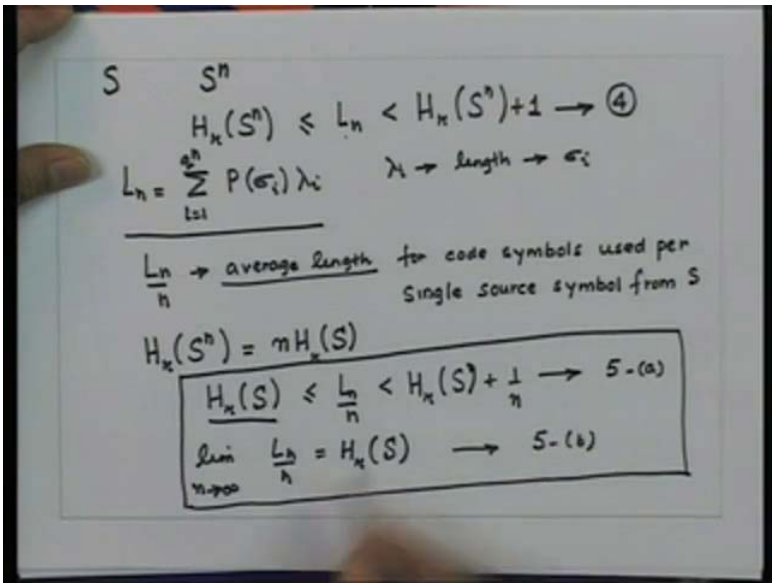
So, taking exponential of the left inequality of equation 1 we will get 1 by P i is less than equal to r of Li which implies that P i is greater than r minus Li. Now, summing to over all i we obtain if I sum this all I assuming that the source is of size Q than I can write this relationship and this is 1 therefore, I get this relationship. Now, this relationship is you know that we have discussed this earlier and we have shown that if this condition is satisfied then it is possible for us to get an instantaneous code for that source. So, this choosing Li according to this relationship given by 1 is acceptable for synthesis of a instantaneous code.

Now, equation 1 defines an acceptable set of Li for an instantaneous code multiplying equation 1 by P i and summing up over all, i we will find if i sum this up P i summation log r 1 by P i i is equal to P i log r 1 by P i P i Li P i log r 1 P i plus P i these are summed over all i we get the relationship as this is the entropy of the source measured in array units this average length of the code, another very important result which we have derived.

Now, there is a difference between this result and the result which is. So, earlier another result which we saw earlier was a H r S is less than equal to L there is a important difference between this relationship and this relationship. This relationship expresses a bound for the average land of a code independent of any particular coding scheme the bound requires only that the code be instantaneous, whereas equation 3 on the other hand is derived by assuming the coding method given by equation 1. So, if I use the coding method described by equation 1 then I get this relationship and this relationship provides both a lower and upper bound on the average length of the code.

Now, since this relationship is valid for any 0 memory source we may apply it to the nth extension of the source. Let us see basically what happens if I apply this kind of coding scheme to an nth extension of a source. So, let us assume that I have a source s which is a 0 memory source and I look at its nth extension of this source. If I follow the strategy for coding which is given by this equation then this equation is also valid for the nth extension because nth extension of a 0 memory source is again a 0 memory source.

(Refer Slide Time: 18:09)



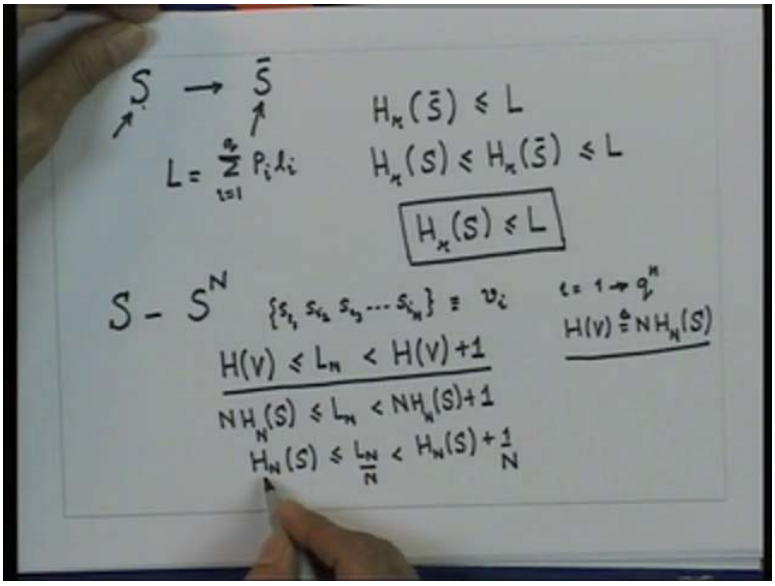
So, the relationship which I get is H r S n is less than equal to a L n less than H r S n plus 1, where L n is the average length which is defined as probability of sigma i lambda i, i equal to 1 to Q where Q is the size of the source Q n Q raise to n is the size of the nth extension source sigma i are the symbols of the nth extension of the source s n lambda i are the length.

So, lambda i corresponds to the length of the code word corresponding to symbols from nth extension that is sigma i. This is the average length and L n by n will give me the average length for code symbols used per single source symbol from S. Now, we have also seen that H r S n that is entropy of the nth extension of a 0 memory source is equal to n times the entropy of the original source. So, based on this relationship we can write this equation. What this equation says is that it is possible to make L n by n as close as we wish to H r S by coding the nth extension of s rather than original source that is S because S keeps on increasing 1 by n tend towards 0. So, in that case L n by n will tend towards H r S that is entropy of the original source.

Now, sp limit of n tending to infinity would be L n n is equal to H r S. This is a very important relationship which we have derived. This equation is known as Shannon's first theorem or the noiseless coding theorem. It is one of the two major theorems of information theory equation 5 a tells us that we can make the average number of r array code symbols per source symbol as small as possible, but not smaller than the entropy of the source measured in r array units.

So, the prize which we pay for decreasing L n by n quantity is the increasing complexity of the coding scheme. Now, all this discussion which we have done pertains to a 0 memory source and its extension. The next question arises is are this results also valid for a source which is not a 0 memory source, but say a Markov source. Let us extend our discussion to a Markov source.

(Refer Slide Time: 23:59)



Let me assume that I have a Markov source S and have its adjoined S bar. We have seen the definition of a adjoint of a Markov source. Adjoint of a Markov source is a source with the source with the same source symbol as the source alphabet as the source S. And the symbol probabilities of the symbols in s bar is the same as the first order symbol probabilities of source S. So, that is a definition of S bar. Now, the process of encoding the symbols in the source S and the symbols which are identical to the source in the source alphabet S into an instantaneous block code is identical, because the source symbols are same. And the probability of the symbols are also the same.

In that case what it means that the average length which is defined as p i L i i is equal to 1 to Q this average length is also identical for both S and S bar. S bar however is a 0 memory source and we may apply the earlier derived result to obtain H r S bar is less than equal to L. Now, we also have seen that entropy of the original source is always less than or equal to the entropy of its adjoint. And this is less than equal to L. So, what follows is that H r S is less than or equal to L.

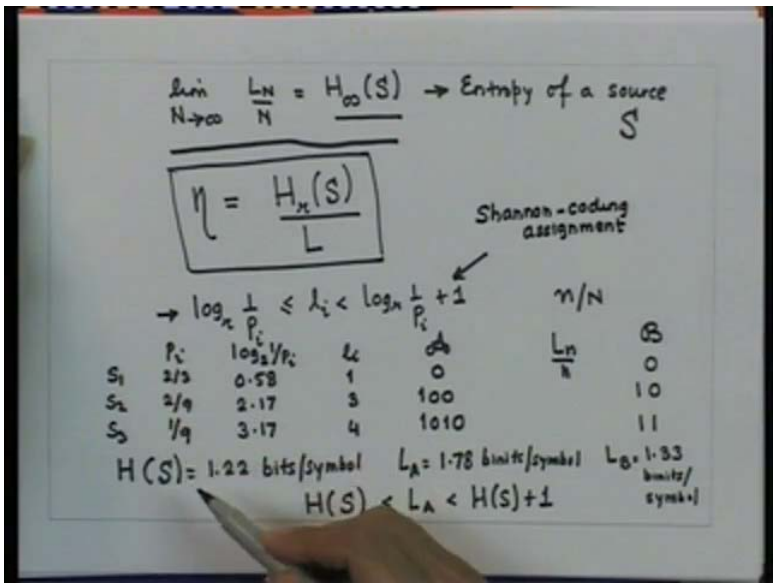
So, again even for a Markov source we have shown that the average length of a code when we code the symbols individually in the source is greater or equal to the entropy of the source. Let us extend this result to an nth extension of a Markov source. So, I have a source S which is a Markov source and I am looking at the coding of this source in groups of n elements. So, that means I am looking at s n where i code Si 1 Si 2 Si 3 up to Si n and as 1 unit.

Now, if I start coding the source original source s in blocks of n source symbols then I can say that this source symbol form a new source symbol which is the i. And now I will range from 1 to 2 raise to n. Now, let us follow the same strategy which we followed earlier for a 0 memory source. Now, for a 0 memory source and its nth extension we had derived a result saying that L n is greater or equal to this is result which we had derived for the 0 memory source.

And it is a nth extension, now based on this same thing is valid for a Markov source I can say that L n is greater than or equal to H of v less than H of v plus 1. So, the same result is valid when I code this source in groups of n source symbols. Now, we have also seen that by definition H v is equal to n times H n S, where H n S is the average information per symbol. So, if you go by this definition which we had seen earlier in our

lectures I can write this expression as n H n S greater or equal to L n. Now, if I divide both the sides by n what I get is this result. Now, this is the average length of code symbols per source symbol of S. Now if I take the limit as n tends to infinity.

(Refer Slide Time: 31:14)



I will get limit n tending to infinity L n by n is equal to H infinity S. And this is the entropy of a source which is a Markov source. So, again we have derived the result that if I want my average length to be as close as possible to the entropy of the source, then I have to use the extensions of the source. Now, another parameter we can define which relates the entropy and the average length of the source and that is known as efficiency of a code. Efficiency of a code is defined as entropy of source divided by the average length of the code for that source. So, this is by definition and efficiency of a code, what is desired is to have as large as possible the value for this eta.

Now, a question that arises is that we have used the coding technique which is based on this inequality. So, this inequality provides a some method of choosing the word length L i by encoding the symbols from S n and taking n sufficiently large the quantity L n by n can be made as small as possible, but it cannot be smaller than the entropy of the source. The question is suppose if n is not very large n or capital N, if this quantity is not very large the theorem does not tell us what value of L what is the average length we shall obtain in that case. It does not guarantee that L or L n by n will be smallest for that fixed n.

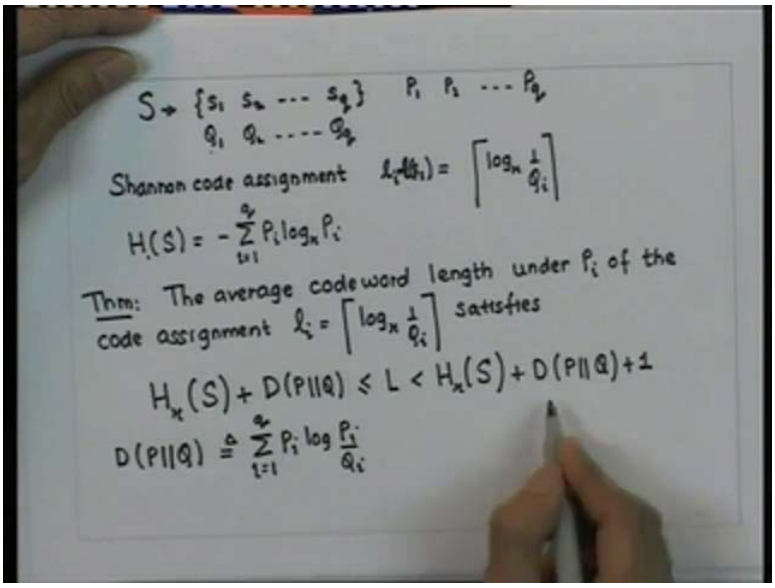
Let us try to understand this with a simple illustration. Suppose, I have a source consisting of 3 symbols S1 S2 S3 and let us assume the P i is for this is given as two third, 2 by 9, 1 by 9 log of 1 by P i. If I assume that I am going to design a binary code than is equal to 0.58, 2.17, 3.17. So, my L i which follows in equality will be 1 3 4. So, I can design a code which is 0 1 0 0 1 0 1 0.

So, I have designed a code a which is based on this coding strategy. Now, if you calculate the entropy for this source turns out to be H s is equal to 1.22 bits per symbol. And if I calculate the length for this code is 1.78 binits per symbol. If you look at 1 .78 it satisfies this inequality of H S this inequality is satisfied by this code a, but unfortunately this coding strategy does not give me a compact code, a code with the length average length of the code to be as small as possible. So, if I take another code in this case if I have another code say B which is 0 1 0 1 one which is again an instantaneous code. And you can calculate the length for this code turns out to be 1.33 binits per symbol.

So, what it means that this procedure does not guarantee a compact code. Now, this code B 1.33 is very close to 1.22. So, what it also implies that by going for an extension of the source I may not able to achieve much. So, what this example demonstrate is that using this coding strategy, it is not essential that we always get a compact code. We could get a compact code when n is very large then when we consider nth extension of a source.

Otherwise we have seen in this example that follow another strategy I could get average length of the code which is smaller than the strategy for given by this inequality. Now, this strategy of coding is known as Shannon coding strategy or Shannon coding assignment for the lengths. Another question which arises is that whenever the design using this strategy based on some probability of symbols given here, but in reality if this probability is not correct than what is the effect on the average length. So, let us answer this question.

(Refer Slide Time: 39:28)



So, what we say is that we have a source s with source symbols given as S1, S2, up to Sq. The real probability of these symbols are P1, P2 up to Pq, but for some reason this is not available and the estimate of this probabilities are available. So, let us call those estimates given as Q1, Q2 up to Q q. So, as far as we are concerned since this is known to us we will be designing our code based on this information.

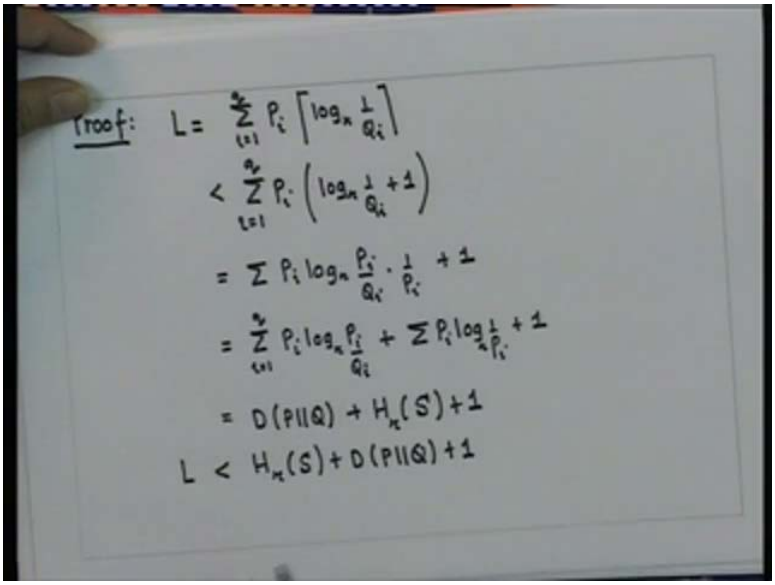
So, when we do that what it means that if we use Shannon code assignment. Then my lengths for the code word for each source symbol will be decided by log of this is a symbol which we use for finding out the first integer larger than this value. So, our designing of L i will be based on Q i. Now, the two function or the two probabilities are given by this. So, the entropy of the source is P i log r P i. So, this is a real entropy of the source.

And if we design our code properly than we expect that every length of our code should be as close as possible to H S, but since this is not known to us and well be designing our code based on the length given by this probabilities. Then the average length of the code which we will be getting in a practical case would be very much different from the entropy of the source.

How much is the difference is in the entropy and the average length based on this is of interest to me. And this difference is given in the form of a theorem. The theorem says that the average code word length under real probability of real probabilities given by P i

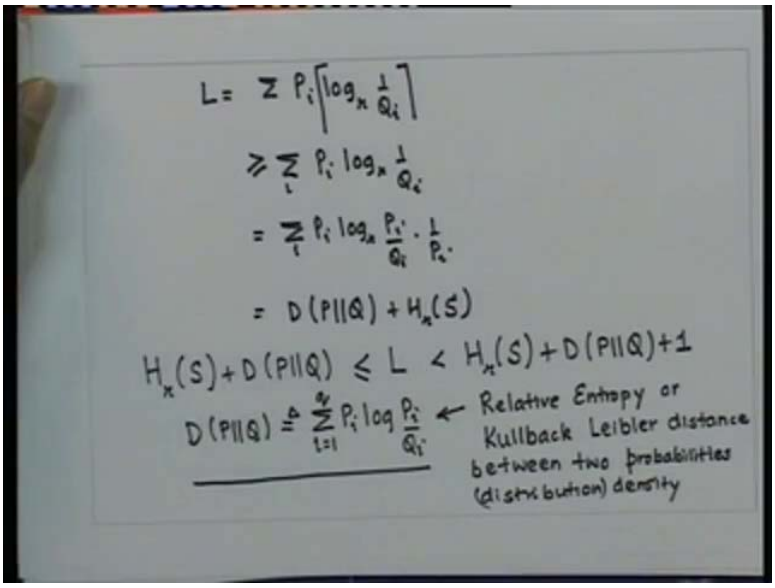
of the code assignment L i equal to log of 1 by Q i. This is what is available to me satisfies the following relationship where by definition P of Q is by definition given as P i log of P i Q i i is equal to 1 to q. So, this is what we want to prove.

(Refer Slide Time: 44:33)



Let us look at the proof of it the average length of the code which we will get is L is equal to P i is the real probability of occurrence of a source symbol. And the design length for this source symbol S i would be given by this quantity because Q i are available to us. And this quantity is less than P i i equal to 1 to Q log of r 1 by Q i plus 1 this is equal to P i log of P i Q i 1 year 1 by P i 1. This is equal to summation of P i log r P i Q i plus P i log. This by definition is equal to d of plus H r S plus 1. So, we have shown that L is less than H r S plus. Similarly, we can show the other side of the inequality not very difficult to do that.

(Refer Slide Time: 47:12)



So, we write L is equal to a again P i log of 1 by Q i and this is greater or equal to summation where i P i log r 1 by Q i is equal to this can be shown as equal to. So, finally, we get result as H r S plus d of Q where d of P Q is by definition given as P i log P i Q i. So, this is known as this is termed as relative entropy or it is also known Kullback Leibler distance between two probability distributions. More appropriate would be to say to two probabilities density.

So, the penalty which we pay for the wrong choice for the probabilities of the source symbol is in terms of the relative entropy. So, our average length is going to increase by this quantity. So, we have looked at the mathematical relationship between the average length of a code and the entropy of the source for which the code has been designed. We also look at some of the methods to synthesize an instantaneous code. Now, in the next class we will have a look at different coding strategies which can be adapted in particular scenario to obtain average code word length to be as cool as possible to the entropy of a source without going further extension of the source.