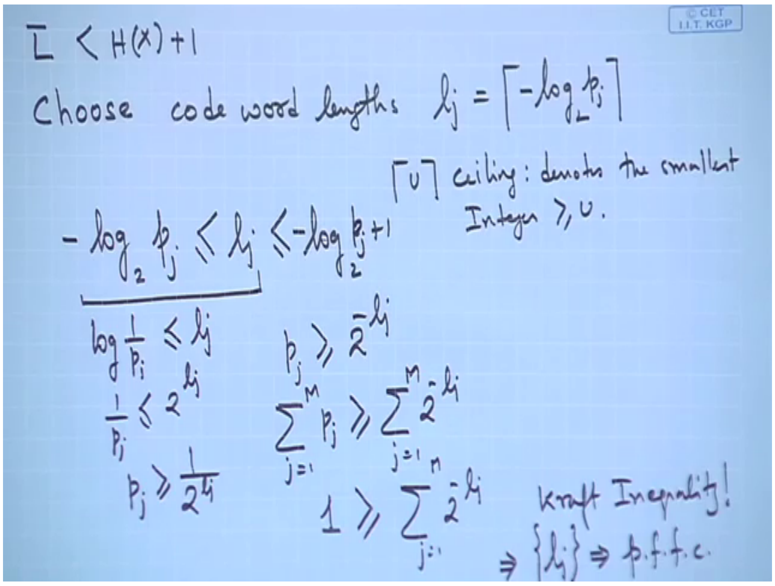
Modern Digital Communication Techniques

Prof. Suvra Sekhar Das
G. S. Sanyal School of Telecommunication
Indian Institute of Technology, Kharagpur

Lecture - 12
Source Coding (Contd.)

We continue in this particular lecture and we discuss the other side of the inequality where we try to see that L bar is less than H x plus 1; that means, the minimum value of the expected code word lengths is at most deviating by one bit more than the entropy.

(Refer Slide Time: 00:50)



$\bar{L} < H(X)+1$

Choose code word lengths $l_j = \lceil -\log_2 p_j \rceil$

$\lceil u \rceil$ ceiling: denotes the smallest Integer $\geq u$.

$$\underbrace{-\log_2 p_j \leq l_j}\leq -\log_2 p_j + 1$$

$$\log \frac{1}{p_i} \leq l_j \qquad p_j \geq 2^{-l_j}$$

$$\frac{1}{p_j} \leq 2^{l_j} \qquad \sum_{j=1}^{M} p_j \geq \sum_{j=1}^{M} 2^{-l_j}$$

$$p_j \geq \frac{1}{2^{l_j}} \qquad 1 \geq \sum_{j=1}^{n} 2^{-l_j}$$

Kraft Inequality!

$\Rightarrow \{l_j\} \Rightarrow$ p.f.f.c.

So, we move further and what we say is that we would like to show that prefix free codes exist for l bars with H x less than 1. So, if we can show that the prefix free codes exists for this particular set up then we can of course, say that since L bar min is less than or equal to L bar then of course, we can find prefix free codes which satisfies this particular condition; do so, we would say that is start of by choosing the code word lengths l j which is equal to the ceiling of minus log base 2 of p j. So, where we could write this ceiling function it denotes the smallest integer greater than or equal to u right that is well known. So, what we know is that we can write that minus log base 2 of p j from this we can write for l j lies between this on the lower side and on the upper side, minus log base 2 p j plus 1.

So; that means, from this it is either equal to this or greater than this, and it is less than this particular number. Now when a p j is an integer number integer power of 2 of course, this holds with equality and of course, this is also satisfied. So, when we look at this particular part of the inequality, we can write that log of 1 by p j is less than or equal to l j.

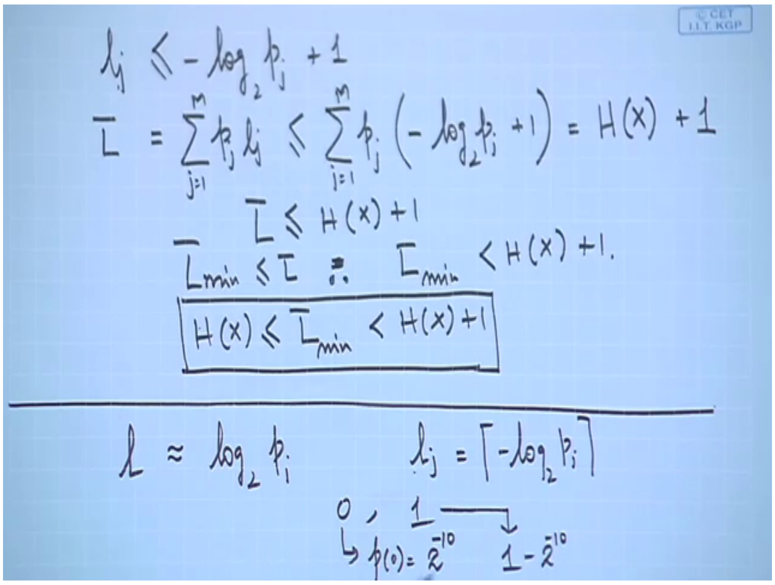
And therefore, we can say that 1 upon p j is less than or equal to 2 to the power of l j or in other words that would also surprise to mean that p j is greater than or equal to 1 upon 2 to the power of l j or you could also say that p j is greater than or equal to 2 to the power of minus l j right. So, we have arrived at this point from this definition and if you take summation over l j on both the sides, what you have is summation of j from 1 to m p of j is greater than or equal to summation of j from 1 to m, 2 to the power of minus l j. So, this since it is a probability space is one and this side is j equals 1 to m 2 to the power of minus l j. So, what it shows is that these code word lengths if shows an according to this satisfies the Kraft inequality.

Now, since it satisfies the Kraft inequality what it employees is that, these set of lengths that we have started with we started with these set of lengths it can construct prefix free code words. So that means, if we take this kind of code word lengths which we have selected from the previous solution where we have minimized the code word lengths by taking the Lagrangian and hence taking the derivative we have found that is l j is equal to minus log of p j or log base 2 one upon p j, and that was for real lengths

And then we said if we have to take integer lengths you can take these ceiling functions which are the nearest integer. So, even by selecting that we can say that one can find these codes set of code word lengths would satisfy the Kraft inequality; that means, you can conceive you can construct prefix free codes. So, that is the first constrained that it satisfies, now we move down further and we try to see what do we have with us now.

So, in other words what we have to summaries that you can construct prefix free codes with this alright. So, moving further.

(Refer Slide Time: 05:22)



$$l_j \leqslant -\log_2 p_j + 1$$

$$\bar{L} = \sum_{j=1}^{m} p_j l_j \leqslant \sum_{j=1}^{m} p_j \left(-\log_2 p_j + 1\right) = H(X) + 1$$

$$\bar{L} \leqslant H(X) + 1$$

$$\bar{L}_{min} \leqslant \bar{L} \quad \therefore \quad \bar{L}_{min} < H(X) + 1.$$

$$\boxed{H(X) \leqslant \bar{L}_{min} < H(X) + 1}$$

$$l \approx \log_2 p_i \qquad l_j = \lceil -\log_2 p_i \rceil$$

$$0, \; 1 \longrightarrow 1 - 2^{-10}$$

$$\hookrightarrow p(0) = 2^{-10}$$

So, if you take the RHS of this; that means, l j is less than or equal to minus log base 2 p j plus 1, since we have this then we can say that L bar which is the expected value of code word lengths by definition it is sum of p j times l j. So, since l j is less than or equal to this you can say that this is less than or equal to sum of p j of course, this is sum over j equals to 1 to m I am not writing it because its repetitive minus log base 2 of p j plus 1. See if you take this particular summation what do you have is sum over p j log base 2 p j with a minus sign which is H x of the entropy, as per definition plus summation of p j multiplied by one that is summation over p j over the entire set is equal to one.

And hence what we can say is that L bar is definitely less than or equal to H x plus 1 simply by taking the right hand side of this, and that because we have taken code word lengths of this. So, what we are shown is by taking code word lengths which are integer nearest integer to the solution of the Lagrangian, you can find code word lengths which satisfy the Kraft inequality that is by taking the left hand side of this and by taking the right hand side of this we have come up to the point where we say that the expected code word lengths are at most one bit more than the entropy.

Now, since L bar min is than less than or equal to L bar therefore, we can clearly write that L bar min is less than H x plus 1 and hence we have the full set of thing that is H x is less than or equal to L bar min which is less than H x plus 1. So, this establishes the entropy bound on L bar min so that means, what we can say in summary is that if you choose a code word lengths, according to this result which was due to the Lagrangian that we have derived in the previous lecture that you take the Lagrangian of the summation and with the constraint that it satisfies the prefix free code, that in the Kraft inequality we ended up with a solution without the ceiling function.

So, when you ended up with this particular solution without the ceiling function, you have real solutions real lengths and we said when these probabilities are integer power of 2 you have the exact solution for which it is the minimum. Now when it is not integer power of 2 you can take the ceiling function to take the nearest integer which is greater than this this particular number, say even by taking the solution we can find that these lengths would satisfy Kraft inequality as we have seen over here and not only that those lengths would produce an L bar which is at most one bit close to H x, and that is because of this right hand side.

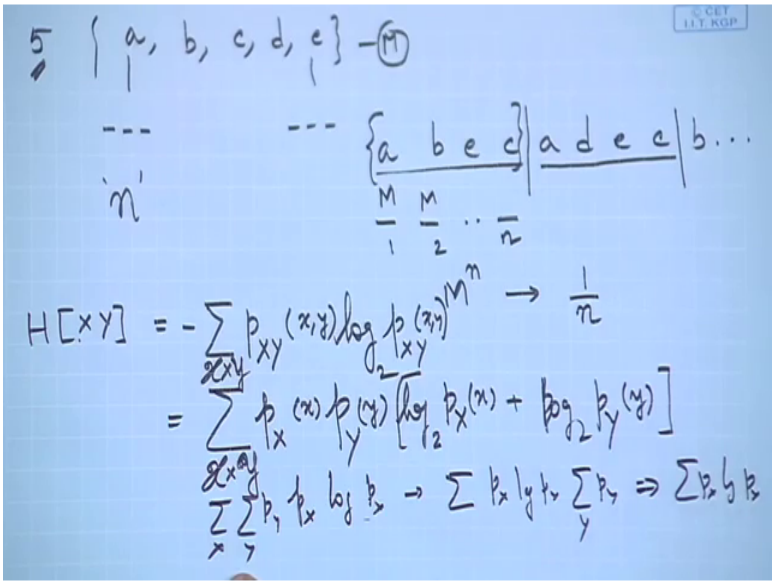
So, all it tells us is that we can select these set of lengths given a set of probability and once you do that you have already found an L bar which is the best solution of the average code word lengths that you need to choose. So, in other words you can say that l is approximately equal to log base 2 of p j or a better way to say is that this length j is equal to the ceiling function of minus log base 2 of p j. Say if it is an integer power of 2 l j is equal to the number inside the ceiling function if it not a integer power of 2 then it is the smallest integer lesser than this.

So, what they there is of course, one particular example which you can take to show the things are not a quite so straight forward, will just do it just for the sake of example suppose I have zeros and ones once coming out, and let us see I have the probability of a 0 being equal to 2 to the power of minus 10 and probability of 1 it is 1 minus 2 to the power of minus 10.

So, going by this we find that the number of bits I should put to 0 is 10,and according to this will be the number of bits that is that I should give to one and definitely what we see is that this is not the one which is going to minimize the average code word length. So, why is it so and what are the things follows out of it and what can be done even to make it very close to entropy is something what we are going to see very soon.

So, before we proceed with the particular details of example, I would like to remind that when we did prefix when we did fixed length codes which we had taken a set of source symbols let us say there be 5 source symbols for example, a, b, c, d and e if I have to give fixed length codes we said that since this is 5, 2 bits would not help because 2 bits would create 4 options. So, you need to take 3 bits. So, every symbol are going to get 3 bits right.

(Refer Slide Time: 11:19)



5 $\{a, b, c, d, e\}$ – $\boxed{M}$

'n' $\{a\ b\ e\ c\}|a\ d\ e\ c|b\ldots$

$\underset{1}{\overset{M}{-}}\ \underset{2}{\overset{M}{-}} \cdots \underset{n}{-}$ $M^n \rightarrow \frac{1}{n}$

$$H[XY] = -\sum_{x,y} p_{XY}(x,y)\log_2 p_{XY}(x,y)$$

$$= \sum_{x,y} p_X(x)\,p_Y(y)\left[\log_2 p_X(x) + \log_2 p_Y(y)\right]$$

$$\sum_x \sum_y p_y\, p_x \log p_x \rightarrow \sum p_x \log p_x \sum_y p_y \Rightarrow \sum p_x \log p_x$$

So, if you do this you would be not doing it very efficiently because with 3 bits you can encoder 8 symbols, but here you have 5 symbols. So, there is a bit of wastage and then we found that probably you can make things better if from the source which produces a sequence and so on instead of encoding one symbol at a time. If we group a few symbols let us say we have taken for groups and we create a super symbol so that means, if I select n symbols in that case if I select n symbols, and if there are m symbols in the consolation.

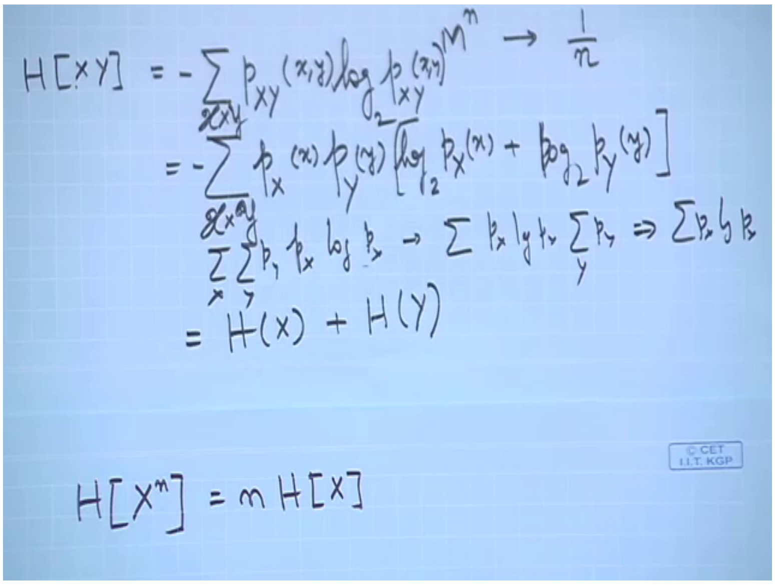
So, what I have is 1 2 dot dot up to n; that means, each position can be taken in m different ways. So, overall if I group these symbols I can say that this is a new symbol and then we can ask how many possible new symbols we have and the answer is m rest to the power of n and now if we encode this particular symbol what we can say is that we can be 1 by n close to the best possible situation. So, by making n very very large you can reduce the overhead in this case or the losses in this case: so will extend that same philosophy for variable lengths codes.

However when you have to do it for variable length code a one have to use the calculus of entropy. So, what we are going to do is we are going to show the expression of H x y. So, if you have H of x y where x and y are 2 different random variables. So, this could be defined as sum over P over the joint distribution of x and y times x log of p x y of x y by the same definition and of course, it is log base 2 and if x and y are independent. So, if x and y are independent.

So, remember we have taken discreet memory less source. So, when we have taken discreet memory less source the in one of the postulates we said that the outputs are independent and identically distributed so that means, 2 symbols that come out do not have memory. So, that was the memory less source. So, here we are talking about the sequence which takes 2 symbols, in this case if it is I I d we can say they are independent.

So, this joint distribution becomes a product distribution and of course, this is over the set of x and y. So, why take the Cartesian product for all combinations of x and y or over the whole set of x and the whole set of y. So, what we have is p of x times x times p of y of y log base 2, p x of x this is multiplication inside the log you have this summation right. So, if you take this first product what you have is p x times this times p y, and since this is over x times y you would first add up through all of y and what we have is p x I will write in short log p x, here p y summation over y summation over x.

(Refer Slide Time: 15:49)



$$H[XY] = -\sum_{\mathcal{X}\mathcal{Y}} p_{XY}(x,y)\log_2 p_{XY}(x,y) \quad M^n \rightarrow \frac{1}{n}$$

$$= -\sum_{\mathcal{X}\times\mathcal{Y}} p_X(x)\,p_Y(y)\left[\log_2 p_X(x) + \log_2 p_Y(y)\right]$$

$$\sum_x\sum_y p_y\, p_x \log p_x \rightarrow \sum p_x \log p_x \sum_y p_y \Rightarrow \sum p_x \log p_x$$

$$= H(X) + H(Y)$$

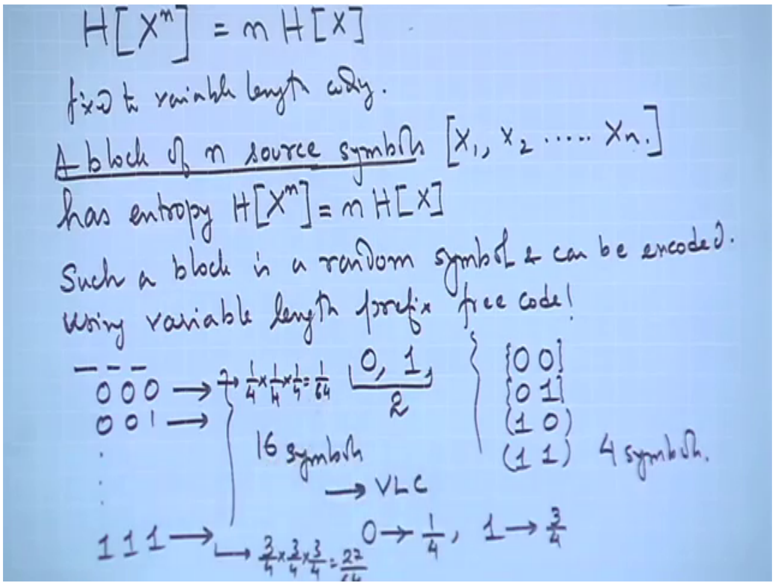
$$H[X^n] = n\,H[X]$$

So, when this is added over all y since these are independent, what this leads to is summation over x log of p x times summation over p y over all of y which would lead to summation of p x log p x and this would equal to one right. So, use that and what we have is this is equal to of course, there is this minus sign H of x, similarly when we apply to this the summation over x becomes one and we are left with a white terms. So, get H of y.

So, for independent sources in general one could write that H of X n is equal to n of H of x; because if we now take x x X n x or x 1, x 2, x 3, x 4 and we call it x, x to the power of 4 because you have taken 4 symbols we are going to get H of x or H of x 1 plus H of x 2 plus H of x 3 plus H of x 4 and if all of them are identical what we have is H of x plus H of x plus H of x plus H of x. So, what we have is 4 times H of x.

Now, if we generalize it and you say that I want entropy of x n; that means, n length sequence where all of these are independent we can write it is H of x plus it is H of x plus H of x H of x plus n times instantly you can write n of H of x. So, since you have now got this.

(Refer Slide Time: 17:51)



$H[X^n] = n H[X]$

fixed to variable length coding.

A block of n source symbols $[X_1, X_2 \ldots\ldots X_n.]$

has entropy $H[X^n] = n H[X]$

Such a block is a random symbol & can be encoded.

Using variable length prefix free code!

000 → ? $\frac{1}{4}\times\frac{1}{4}\times\frac{1}{4}=\frac{1}{64}$
$\underbrace{0, 1,}_{2}$
001 →
16 symbols
→ VLC
111 → $\frac{3}{4}\times\frac{3}{4}\times\frac{3}{4}=\frac{27}{64}$
$0 \rightarrow \frac{1}{4}, \; 1 \rightarrow \frac{3}{4}$
(0 0)
(0 1)
(1 0)
(1 1)
4 symbols.

So, will now try to define what we what we are trying to intend at by or rather it is also known as what we are trying to do is fixed to variable length coding; fixed variable length coding you can take it under this particular heading as well. So, what we have is a block of n source symbols X 1, X 2 up to X n. So, we have this and you say that this has entropy H of X n and since we have seen that H of X n is equal to n H of X n we can say that such a block is a random symbol and can be encoded using variable length prefix free code. So, what we have is a block of n symbols source symbols. So, now, we are selecting this block of symbols.

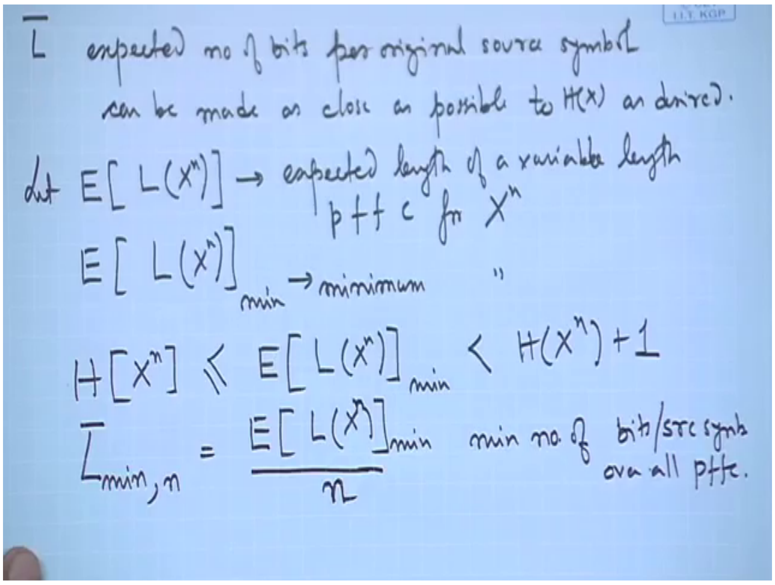
So, it is a fixed length and then missing such a block can be random symbol which we have already explained for example, if you are taking a very basic source as zero and one instead of taking one symbol at a time encoding a zero and encoding a one, you can think of encoding a sequence and let us say 2 bits and we take whenever there is a zero I would encode it as a certain symbol whenever there is 0 1 I would encoded let us at a certain symbol and so on and so, forth. So, I would take 0 0 as my new symbol I would take zero one as my new symbol and so, goes 1 0 and 11.

So, in this new case we have 4 symbols, and we would like to de encode this these 4 symbols which have come from the original source. You could have taken 3 bits and in that case 0 0 0 could be one symbol 0 0 1 could represent another symbol and 1 1 1 could represent another symbol. So; that means, that we now have a source which has 16 symbols previously there was a source which had only 2 symbols and now what we are trying to do is encode this 16 symbols into variable length coding.

Now the result is more or less well understood at this point what we can do is suppose I gave probabilities of zeros and probabilities of 1. So, for example, if I say that zero comes with the probability of let us say a 1 upon 4, 1 comes with the probability of 3 upon 4. So, when I have a 0 0 0 I can calculate the probability as 1 by 4 multiplied by 1 by 4 multiplied 1 by 4 or in other words one upon 64 as a probability of this particular symbol, and the probability of this particular symbol is 3 upon 4 times 3 upon 4 times 3 upon 4.

So, this has a probability of 3 by 4 multiplied by 3 by 4 multiplied by 3 by 4 which is 7, 164 and this is a probability of 1 by 4 times 1 by 4 times one by 4 which is 1 upon 64. So, now, since we have this we can recollect it that we can use lengths which is equal to log base 2 of one upon p j, and take the ceiling function of that and that will give me the code word lengths which to be used to generate prefix free codes which would be minimum possible lengths.

(Refer Slide Time: 22:40)



$\bar{L}$ expected no of bits per original source symbol can be made as close as possible to $H(X)$ as desired.

Let $E[L(X^n)] \rightarrow$ expected length of a variable length p f f c for X^n

$E[L(X^n)]_{min} \rightarrow$ minimum "

$$H[X^n] \leqslant E[L(X^n)]_{min} < H(X^n) + 1$$

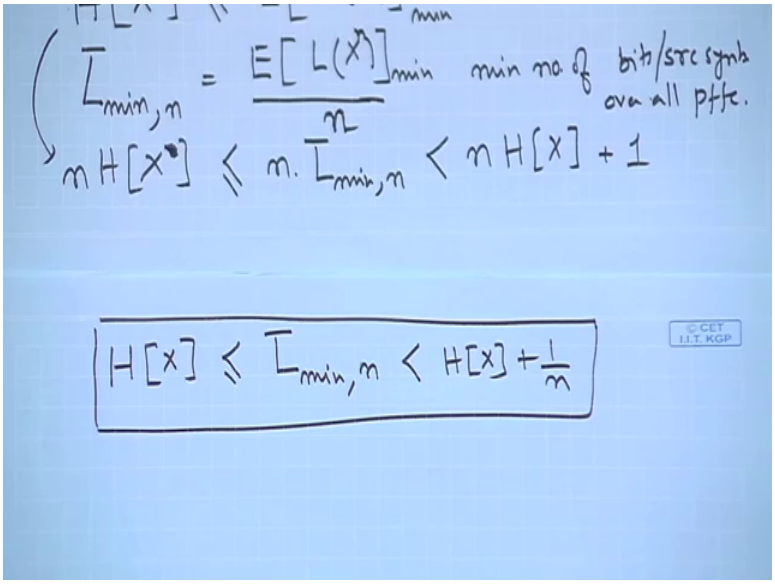
$$\bar{L}_{min,n} = \frac{E[L(X^n)]_{min}}{n}$$ min no. of bits/src symb over all pffc.

So, you can keep doing this until the point or as I make an large is the new result that that is what we are interested in in this particular a discussion.

So, here of course, what we say is L bar is the expected number of bits per original source symbol and what we can say is that this can be made as close as possible. So, this can be made as close as possible to H x this is what we are trying to see as desired. We have seen such a similar result in case of fixed length code we will see the similar thing. So, here we say that let E of L of X n this is to be understood as X n is the block of length n L is the length of code words where the source symbol is made up of n original source symbols and e is the expectation of operator. So, what we can write is this is the expected length of a variable length prefix free code for X n and E of L X n min is naturally the minimum expected length of the same.

So, what we have is that from the previous entropy bound we can say H X n is the lower bund on E of L X n min which is less than H of X n plus 1 this is well known. Now if I define that L min n or L bar min n to indicate that this is equal to E of L X n min upon n; that means, the minimum number of bits per original source symbol over all prefix free codes; that means, look at this this is the minimum expected length of code word for this block. So, since this block is created by n symbols this divided by n would indicate the number of bits per original source symbol right. So, what we have at this point is we extend.

(Refer Slide Time: 26:02)



$$\bar{L}_{min,n} = \frac{E[L(X^n)]_{min}}{n}$$ min no. of bits/src symb over all pfx.

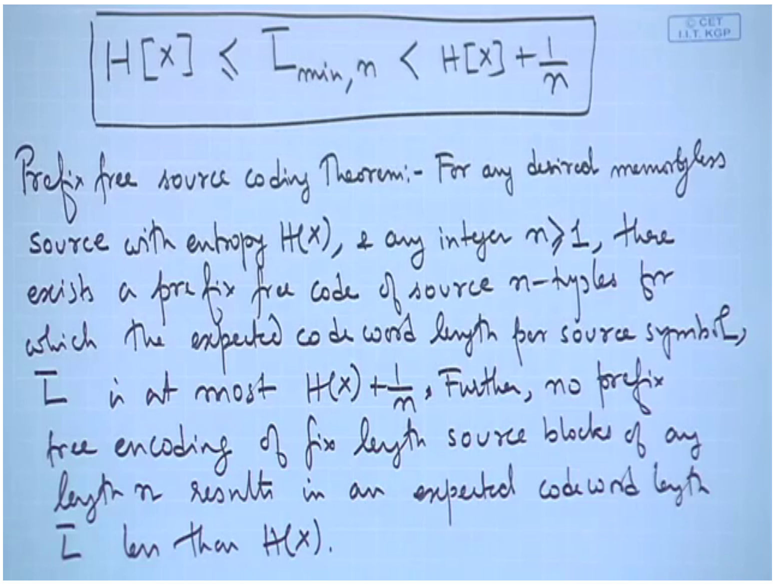
$$n H[X] \leqslant n.\bar{L}_{min,n} < n H[X] + 1$$

$$\boxed{H[X] \leqslant \bar{L}_{min,n} < H[X] + \frac{1}{n}}$$

We go ahead with this and we have this result is already established n H n. So, instead of this we write this and here this is equal to n times L bar min, when taking n tuples at a time is less than again the same result we would apply here n H x this is the mistake which should not be here plus. So, clearly what we have from this if we divide by n all over what we get is H of x is less than or equal to l min bar when taken in a group of n tuples is less than or equal to H x plus 1 upon n.

Now this is a very very important result what we have arrived at which tells us that, by choosing appropriate lengths you can by choosing appropriate lengths; however, if you group n tuples let me instead of taking one symbol at a time if you take n symbols you can make the minimum code word length per original code original source symbol as close as possible to H x, which is the entropy which is the lower limit as desired by taking n very very large and as n tends to infinity this number becomes very close to H x.

(Refer Slide Time: 27:42)



$$\boxed{H[x] \leq \bar{L}_{min,n} < H[x] + \frac{1}{n}}$$

Prefix free source coding Theorem:- For any desired memoryless source with entropy $H(x)$, & any integer $n \geq 1$, there exist a prefix free code of source n-tuples for which the expected code word length per source symbol, $\bar{L}$ is at most $H(x) + \frac{1}{n}$, Further, no prefix free encoding of fix length source blocks of any length n results in an expected codeword length $\bar{L}$ less than $H(x)$.

So, finally, we can state the prefix free source coding theorem as for any desired memory less source with entropy H x and any integer n greater than or equal to 1 there exists a prefix free code or prefix free encoding of source n tuples for which the expected code word length per source symbol L bar denoted by L bar is at most H x plus 1 upon n can also state further no prefix free encoding of fixed length source block of any length n, results in an expected code word length L bar less than H x.

This is the very strong result which tells us that you can design code word lengths such that it is very much possible to bring it as close to entropy as desired because entropy is the lowest limit to which you can achieve and to make things best you can take blocks and these blocks would be if it is pretty large you can make this coding scheme as efficient as possible. We stop this particular discussion at this point we will continue on this topic in the next lecture.

Thank you.