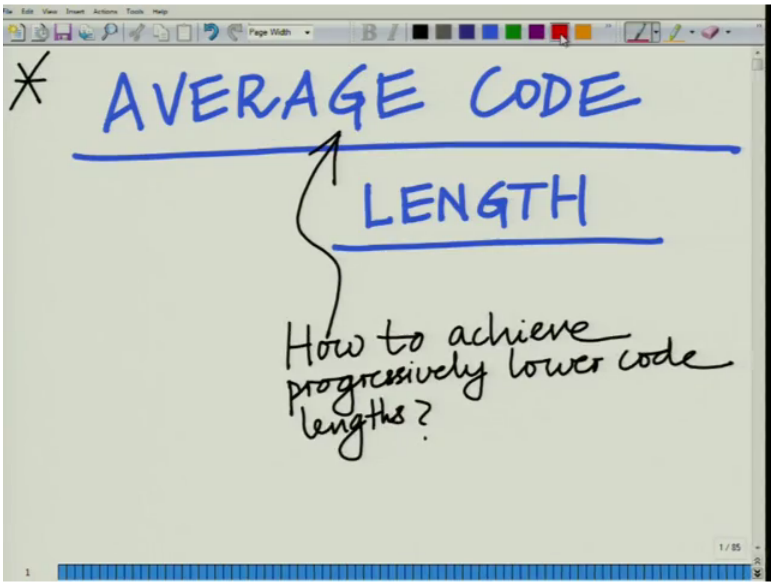
Principles of Communication Systems - Part II
Prof. Aditya K. Jagannatham
Department of Electrical Engineering
Indian Institute of Technology, Kanpur

Lecture – 46
Approaching Lower Bound on Average code length, Block Coding

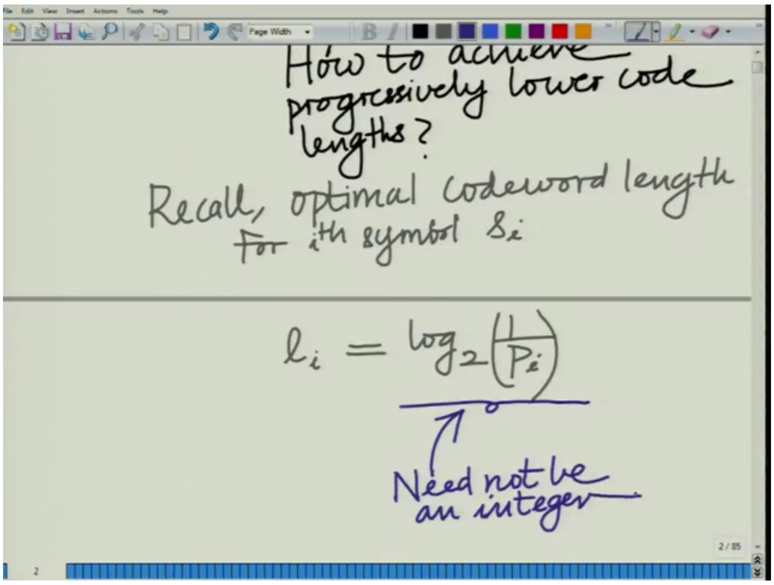
Hello, welcome to another module in this massive open online course. So, we are looking at the average code length and we have seen a very important result where in the average code length is bounded by the entropy of the source and more importantly to achieve this lowest possible average code length, one has to encode symbols such that the length is given by log to the base 2 1 over P i where P i denotes the probability of the i th symbol.

(Refer Slide Time: 00:45)



Now, let us look into this further. So, we would like to look further into this average code length, how to achieve or how to progressively decrease that is what you (Refer Time: 01:08) how to achieve progressively lower code length. So, what do you would like to is would like to design a mechanism, where why we can progressively keep lowering the code lengths moving towards the lowest possible average code length, that is of course, given by the entropy of the source.

(Refer Slide Time: 01:55)

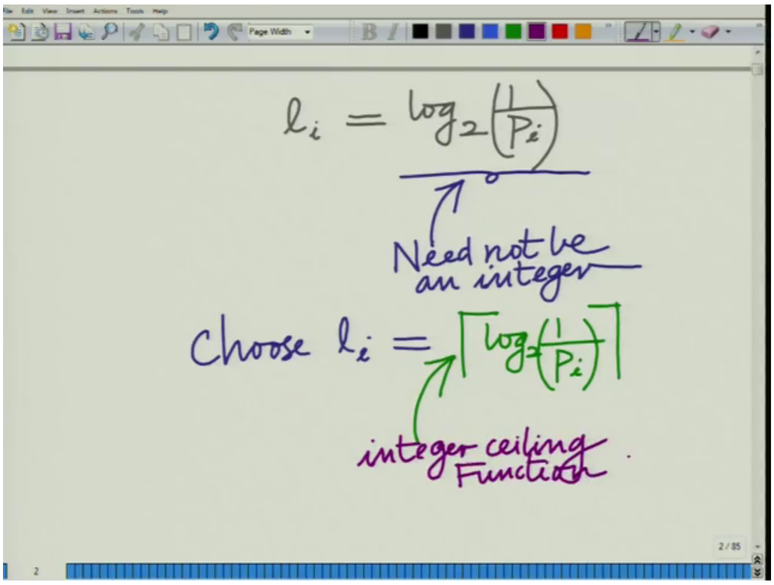


Now, for that what we have seen in the previous module recall optimal codeword length now code length, optimal codeword for symbol i or for symbol s i, remember we had derived this from the constrained optimization framework that is l i equals to log to the base 2, 1 over P i this is the optimal codeword length.

Now; obviously, now we will observe that this need not be an integer that is the problem. So, this need not be an integer which means that of course, when you using binary information bits for instance to represent each symbol then we can only use an integer number of bits, I cannot use half a bit alright. So, I can only represent each symbol using an integer number of bits.

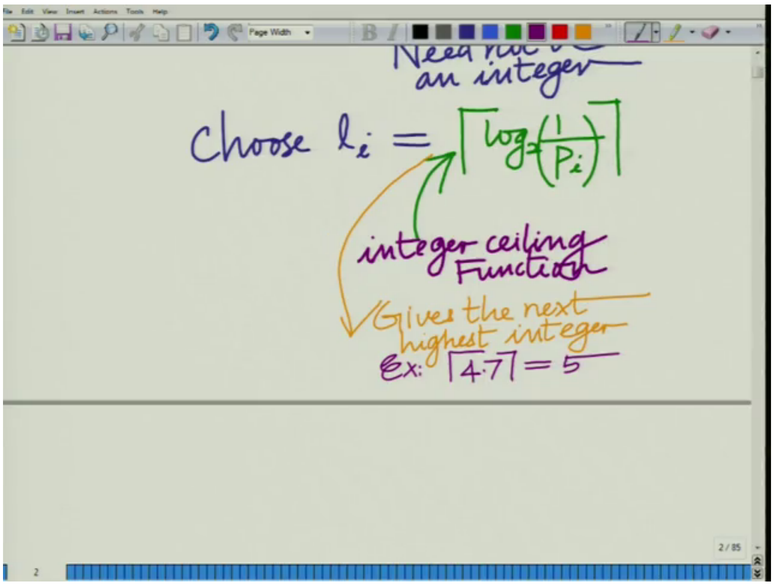
So, therefore, I cannot if this quantity log to the base 2 1 over P i is not an integer then I cannot choose that many bits alright. So, naturally the recourse is to choose the integer which is closest to it now; obviously, I cannot choose an integer which is lower than this for instance if this is 4.5 I cannot choose 4, because then the average code length become would become lower than the (Refer Time: 03:47) it would violet the constraint, but I can always choose an integer which is greater than this. So, I can choose the integer which is the next highest integer corresponding to this, which is given by the ceiling function of this.

(Refer Slide Time: 00:59)



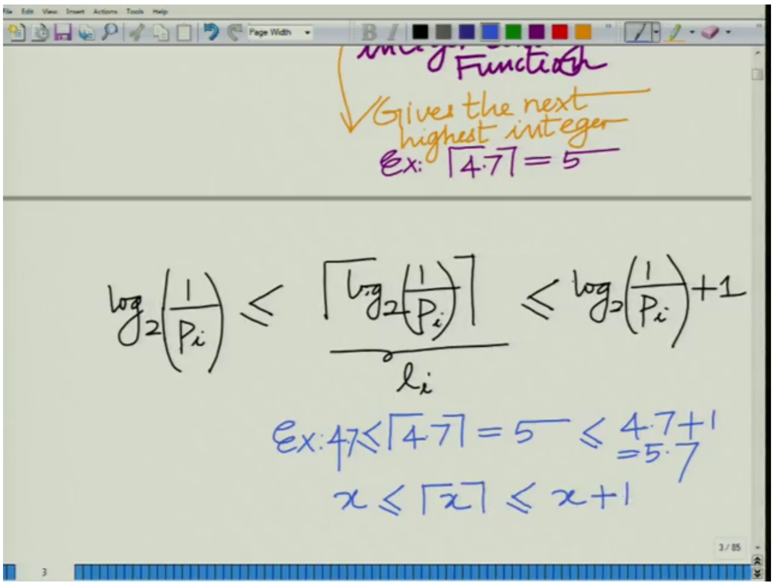
So, I can choose l i equals the ceiling function because if you choose the lower integer then you can violet because you cannot decrease we have a lower bound I cannot decrease it beyond that is I cannot make it in the lower bound I can always choose an average length I can always choose a length which is higher.

(Refer Slide Time: 04:56)



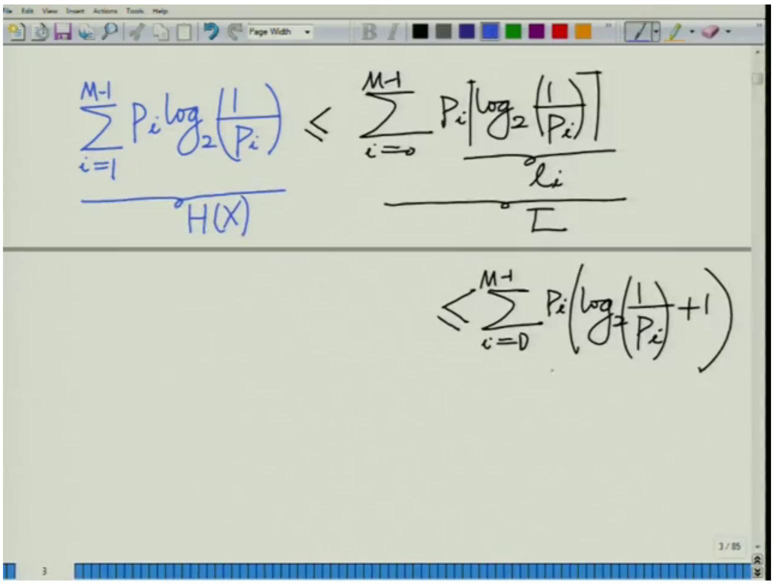
So, this is remember this is the integer ceiling function, this is the integer ceiling function correct gives the next highest integer example for examples ceiling of 4.7 is equal to 5 etcetera so on and it is a very function alright, which you take the next highest integer because it is not an integer you approximated by the next highest integer.

(Refer Slide Time: 05:38)



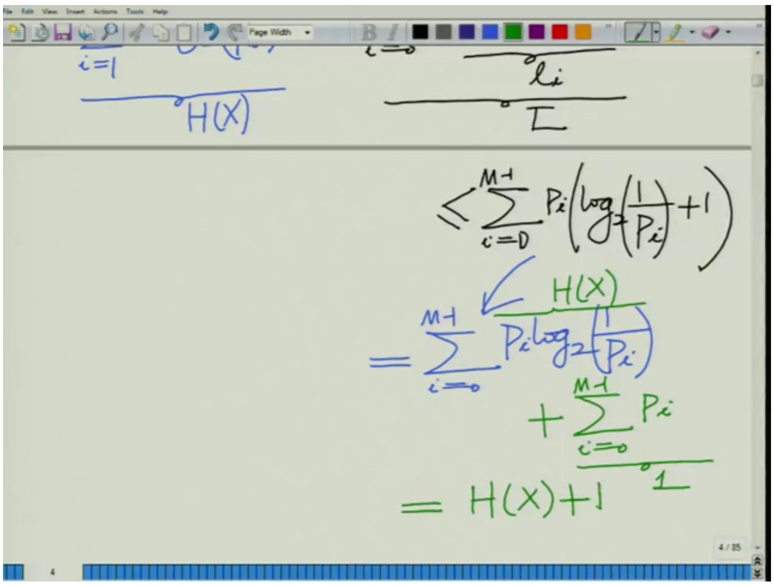
So, therefore, what I have is basically now if you look at this log 2 to the base 1 over P i which is your l i which your setting your l i, it naturally follows that this is less than or equal to log two to the base log to the base 2 1 over P i plus 1 and log to the base 2 1 over P i, and that can be easily seen for example, ceiling of 4.7 is 5 and this is less than or equal to 5 plus 4.7 plus 1 this is less than or equal to 4.7 plus 1 equals 5.7 and this is greater than or equal to 4 and this is greater than and equal to I am sorry 4.7 this is greater than. So, we have inequality that x is less than or equal to ceiling of x, less than or equal to x plus 1, naturally because the ceiling of x is the next highest integer. So, it is greater than equal to x and the next highest integer lies within 1 within a distance of 1 from the given quantity. So, it is obvious less than equal to x by ceiling of x is less than equal to x plus 1 ok.

(Refer Slide Time: 07:25)



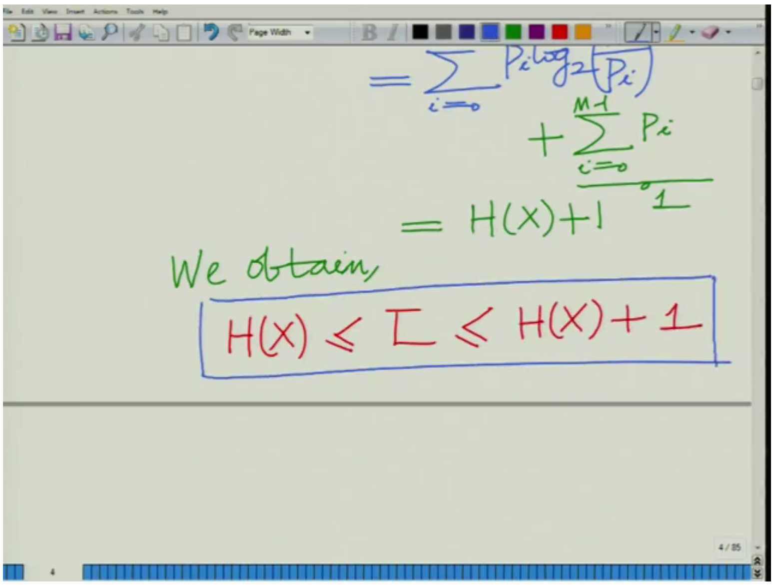
Now, using this now if I multiply on all the three quantities by P i and take the sum. So, I will have P i log to the base 2, 1 over P i summation of i equal to 0 to M minus 1 less than or equal to summation i equal to 0 to M minus 1, P i log to the base 2 1 over P i ceiling which is nothing, but l i and. In fact, if you look at this quantity summation I equal to 0 to M minus 1 P i, l i this is l bar which is less than or equal to summation I equal to 0 to M minus 1, P i log to the base 2, 1 over P i plus 1.

(Refer Slide Time: 08:37)



Now obviously, if you look at this, this quantity is nothing, but the entropy. And this quantity can be simplified as follows this quantity is if you look at this if you take out the, remove the brackets and expand, this is summation i equal to 0 to M minus 1, P i to the be log to the base 2, 1 over P i plus summation i equal to 0 to M minus 1 P i, now this quantity is once again h of x and this quantity is summation P i summation of all the probabilities that is total probability which is equal to 1. So, this is basically H X plus 1.

(Refer Slide Time: 09:20)



And therefore, what you derive net inequality, we obtain what we obtain is basically we obtain the inequalities H X less than and equal to l bar less than and equal to H X plus 1 and this is interesting because which shows that you can obtain using these lengths that is approximating by integers, you can obtain a code which is within one bit of the entropy. So, the lower bound is entropy ideally we would like to achieve that, but observe that the interesting thing about the theorem is basically is C source coding theorem which we derived from the Kraft inequality, is it does not it only gives us a lower bound that is tells us that the average code length is greater than or equal to the entropy however.