Principles of Digital Communications
Prof. Shabbir N. Merchant
Department of Electrical Engineering
Indian Institute of Technology, Bombay

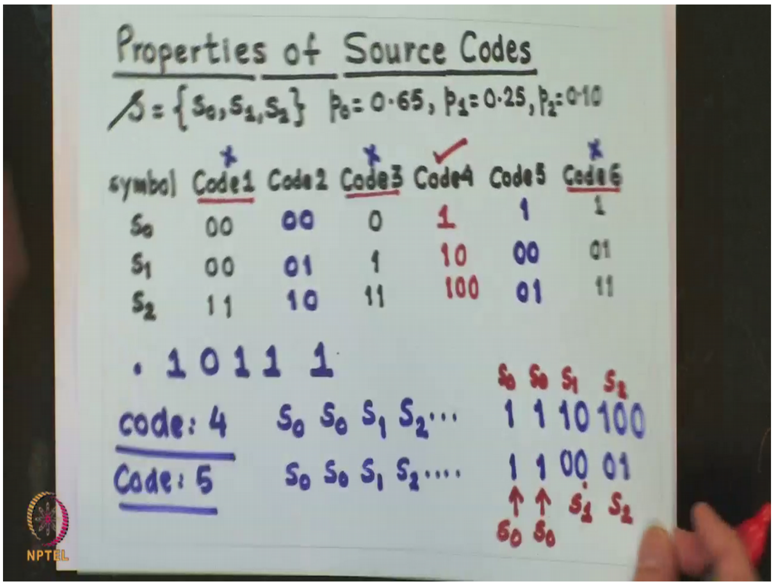
Lecture – 04
Prefix Codes and Kraft's Inequality

So, in the last class we studied the Shannon's first theorem which provides the fundamental limit on the representation of the data which is being generated by information source. What it said that for a discrete memoryless source the average length of the source code which you could have has to be always greater than equal to the entropy of a discrete memoryless source.

And we also mentioned that one of the requirement of a source encoder is to provide uniquely decodable source code. By this we mean that given the output binary stream from the source encoder, I should be able to uniquely decode source symbol sequence from the discrete memoryless source which is input to the source encoder.

Now, today we will study the properties of the source code which will satisfy this requirement of the source code to be uniquely decodable.

(Refer Slide Time: 01:35)



So, let us take an example. I have a source alphabet given by 3 symbols with the probability as specified here and let us I have shown 6 codes which we can design for this source alphabet.

Now, if you take the code 1, we will see that basically there is a problem with this code in the sense that when I receive binary sequence 0 0 I am unable to decode it because there is a confusion by that belongs to S 0, S 1. So, this is not acceptable the first requirement on the source code is that the code words should be distinct. So, this is not acceptable code one because the code one for symbol S 0 is 0 0 and the code word for S 1 is also 0 0. So, code one is ruled out.

Let us take the code number 6 it appears that the code words are distinct, but again there is a problem with this in the sense that if a sequence of binary stream is a valid output I am unable to decode it I am unambiguously. Let us take an example let us take say example I have a binary sequence at the output of the source encoder as follows and if we try to decode and if we know that this has been encoded using code 6 then there is a problem because 1 I can decode it S 0, 0 1 I can decode it at S 1, but this there is a problem I do not know whether it is S 0 followed by S 0 or it is just S 2. So, this is ruled out.

If you look at code 3 again the same problem arises, correct. Though the code words are distinct, but when these code words appear in a sequence it is not possible for me to decode it unambiguously correct. So, this is ruled out. Then let us take. So, this and this are ruled out.

Let us look at the code 2, code 4 and code 5. If you see basically again the code words are distinct. So, by the first requirement this should be acceptable and let us see basically what happens when I get the binary stream at the output of the encoder and see how we are able to decode it. Let me take an example suppose if I were to use code 4, and encode S 0, S 0, S 1 and S 2 I would get the output as 1 1 1 0 1 0 0. If I use code number 4 and if I use code number 5 for the same source symbols sequence I would get this as 1 1 0 0 0 1. Let us see the difference between the two.

If we observe the difference, if you look at the binary stream when I get one here I cannot immediately can I immediately decode this I cannot because I do not know whether this is the end of the code word for the symbol S 0 or it is basically the first symbol of the code word S 1 or S 2. So, I have to wait for the next symbol to arrive in this. So, I will not be able to take the decision, but once this 1s arise I can decide that this is equal to S 0, and then I come to this symbol again I cannot decide unless I have observe this next symbol in this sequence.

So, then I will be able to decide S 0 and then basically when I do this basically I will get this I get this again I will have to wait here and then again I will have to wait here to see whether it belonged to S 2. And only after this has arrived I can decode this as S 1 and finally, I will decode this S 2.

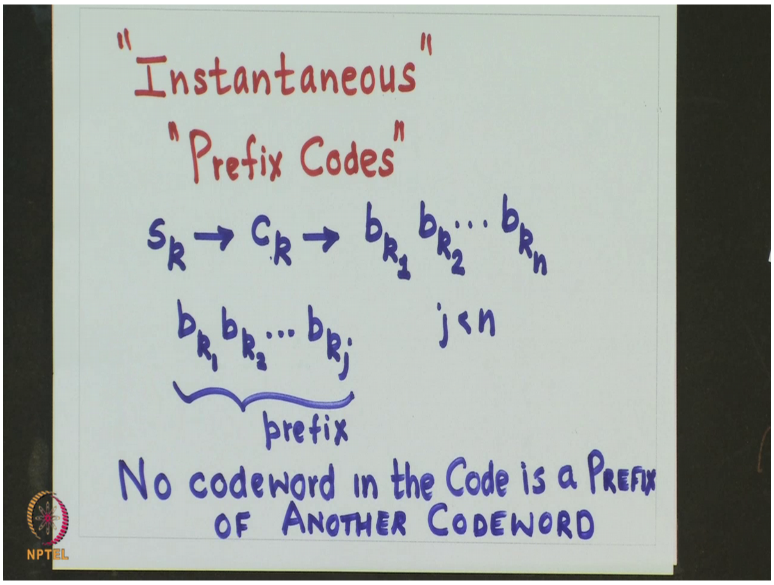
So, what happened basically in this case I need to wait for the future symbol to arrive to decode my present code word. But let us look basically at the same sequence using the code 5 if I do this basically when one arrives I know basically that this is the symbol S 0 when this again here arrives one I know it is symbol 0. When this arrives 0, I know that this cannot be a code word I have to wait for the other thing I am I am sure about that, correct.

So, basically I wait for the next symbol to arrive and I know the length and I am also given the maximum length which a code word has in the code I know S 2. So, once I know this basically at this point itself I decide that this is equal to S 1 and if I look here basically this I know that it is a start of a code word and here basically I know it is a end of the code word and this has to be S 2, correct.

So, you see basically here basically this once they serve as the some kind of a signaling for the end of the code word and the length also helps me to decide the end of the code word. So, this is the difference between code 5 and one of the code 4 correct. And if you see the code 2 also basically I can immediately decide which symbol as arrive after I have observed for two signaling intervals, ok.

So, this code though acceptable 4 it is I have to wait for the future symbol to arrive. So, this is uniquely decodable, but not instantaneous whereas, code 2 and code 5 are known as instantaneous codes. They are also known as prefix codes.

(Refer Slide Time: 08:30)



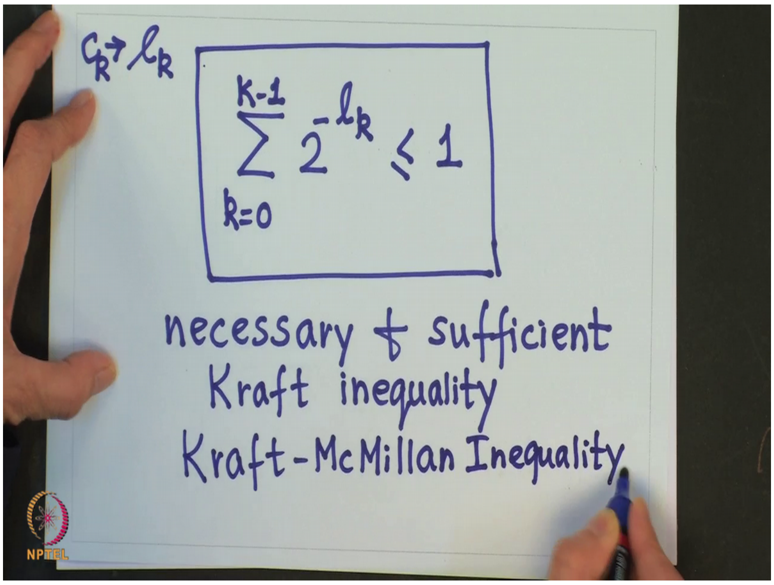
This instantaneous or prefix code have very interesting property. So, what is that property? To understand that let me understand the way the code words are formed.

So, I have a symbol S k and to this basically I have a sign of code word C k and this C k is nothing but a sequence of binary symbols. So, I call b k 1 it could be 0 or 1, then b k 2 the second symbol in the sequence corresponding through the code word C k correct and I have this. So, I am assuming that the length of this code word is equal to n

Now, in this code word b k 1, b k 2 up to b k j for j less than or n, for j less than n this is known as the prefix of the code word, correct. So, b k 1 is the prefix of this code word b k 1, b k 2 is also the prefix of this code word b k 1, b k 2 up to b k 3 is again a prefix of the code word.

Now, one of the property which instantaneous or prefix code should satisfy in order to decode the sequence instantaneously is that no code word in the source code is a prefix of another code word. So, if you have this condition satisfied then that source code is known as a prefix code and once we have a prefix code we will be able to decode the output sequence instantaneously. And there is another important property which all this prefix code satisfied and this condition is basically on the length of such codes.

(Refer Slide Time: 11:30)



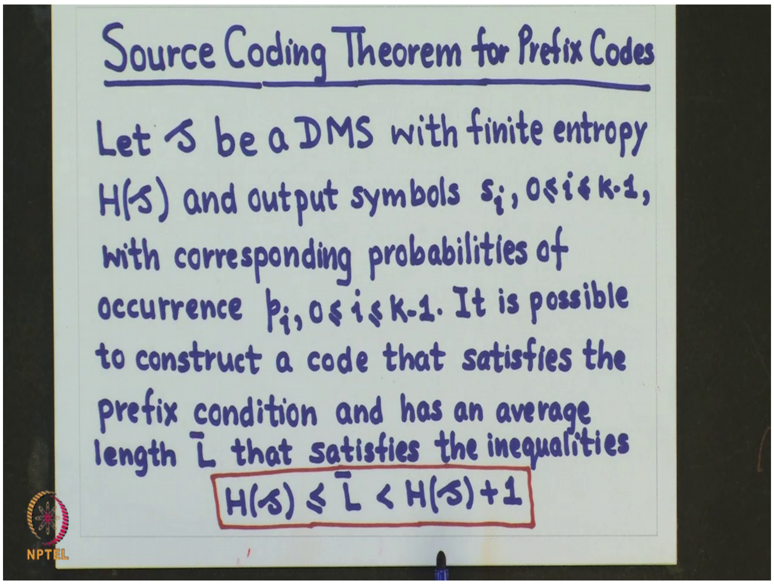
So, we mentioned that this code words have the length l k then we can show that summation of two raised to minus l k, K is equal to from 0 to capital K minus 1 will be always less than equal to 1. It can be shown that this is a necessary and sufficient condition for the existence of a prefix code.

So, this condition is basically on the length of the code correct. In fact, so this is known as Kraft inequality. Remember instantaneous code is a subset of a uniquely decodable code and later on this property was also extended by McMillan to uniquely decodable code. So, for uniquely decodable code also this condition is satisfied. So, sometimes this condition is known in this inequality is known as Kraft-McMillan inequality.

So, what it says that if you take this source then it says that for this source that is take there is a code word are available with length 1, 2 and 2. And you can prove that this will turn out to be, if you just take the, you will get 1 by 2 plus 1 by 4 plus 1 by 4 that turns out to be 1. So, this means the length will satisfy the Kraft-McMillan inequality. So, there should exist a code of length 1 2 2 correct and you see that basically I could design that code in the form here. So, this is the instantaneous codes which I could design, ok.

Now, we will use this Kraft-McMillan inequality to prove the following version of the lossless source coding theorem which applies to codes that satisfy the prefix condition.

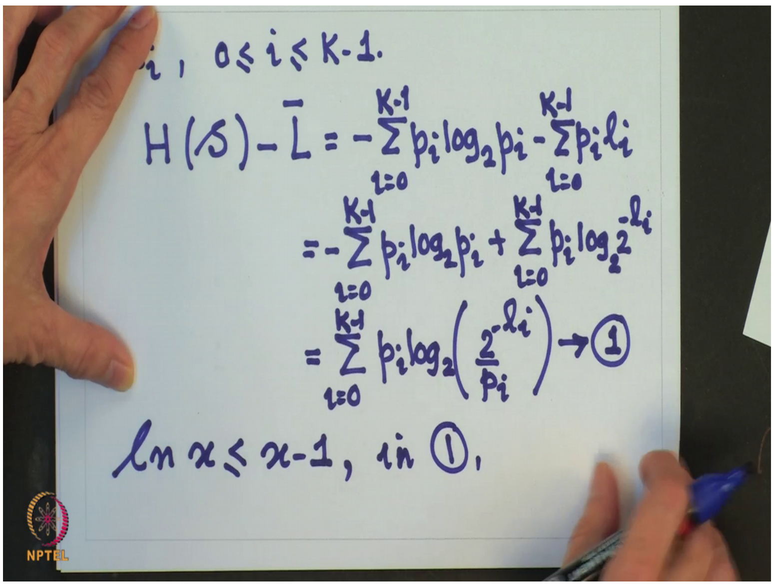
(Refer Slide Time: 14:08)



And what it says is that given a source which is a discrete memoryless source with finite entropy H s and output symbols with the corresponding probabilities it is possible to construct a code that satisfies the prefix condition and has an average length that satisfies the following inequalities, correct. So, let us see how to prove this, correct.

So, to establish the lower bound we note that the code words have the length l i, for i greater than equal to 0 and less than equal to K minus 1.

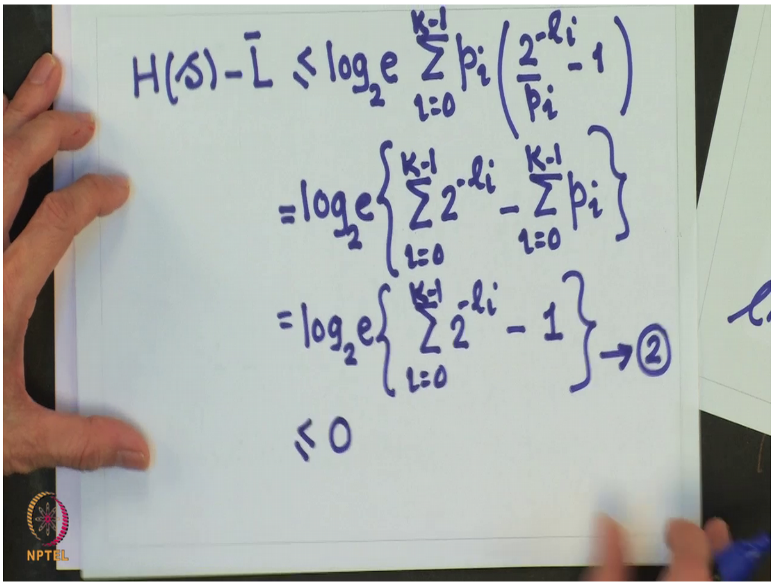
(Refer Slide Time: 14:46)



So, let us take the this difference H s minus l average which is nothing, but minus pi log to the base 2 pi minus pi l i, i is equal to 0 to K minus 1. I can simplify this as follows. I have written l i has minus of log two of the quantity 2 raise to minus l i. So, this I can rewrite as.

Now, we will use the inequality log x less than or equal to x minus 1 for x greater than or equal to 0 in the above equation 1 and if we do that what we get is as follows.

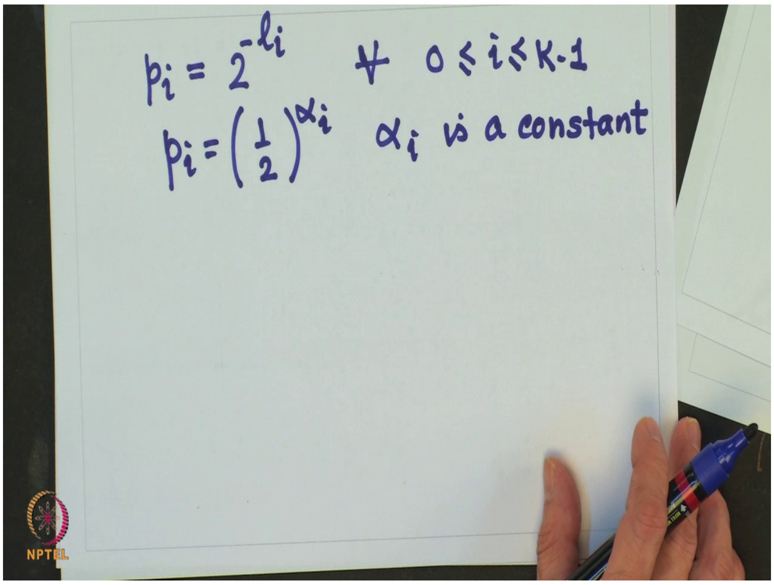
(Refer Slide Time: 16:55)



Now, so from this expression and using the Kraft-McMillan inequality, if the if your source code satisfies Kraft men McMillan equality and this has to be satisfied for uniquely decodable code or for that matter prefix code this quantity will be less than or equal to 1. So, what it implies that this quantity is less than equal to 0, correct. So, this implies that your l bar has to be always larger than H s, ok.

So, now in this case we know that equality will hold only if this quantity is equal to 1 correct.

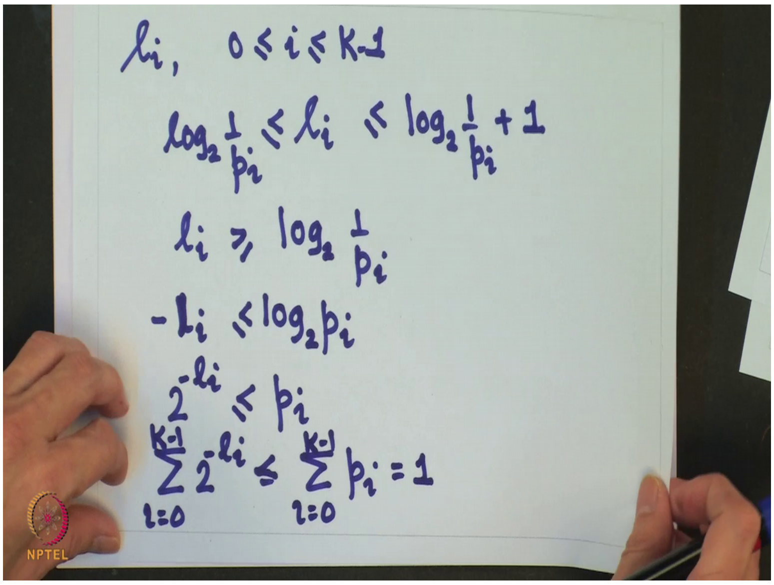
(Refer Slide Time: 18:48)



What will happen is that I should have my p i to be of the form 2 raise to minus l i for all is satisfying this range. So, it means that my p i will be of the form 1 by 2 alpha i, where alpha i is a constant correct. So, if alpha i is a constant and p is of this form then I can choose my l i to be satisfying this equation and then I will get equality correct and that case the average length will turn out to be equal to exactly the entropy of the source.

Now, let us try to establish the upper bound and that can be done as follows.

(Refer Slide Time: 19:46)

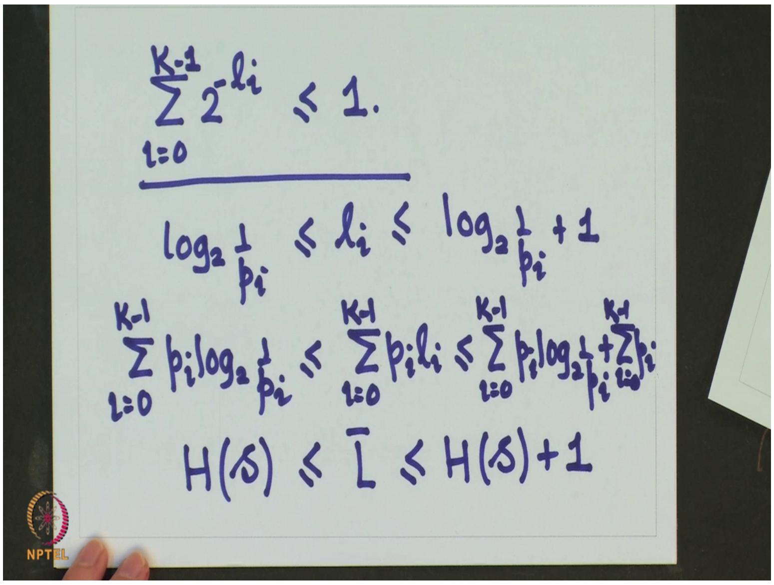


So, we know that this l i's which are the length have to be integers. So, what we do is that we select l i satisfying this constraint, ok. This is because p i is not of the form 1 by 2 raised to some integer. So, if that is not true then you will have to choose your l i in another of manner and this is what we are saying that you choose your l i to be of this form and then we can show that that this will if you take this l i following this inequality then I should be able to generate my prefix code as follows.

So, let us take your l i is larger than equal to log of p i. So, this implies that minus l i is less than equal to log to the base 2 p i. So, this implies that 2 raised to minus l i is less than equal to p i. So, this implies that summation of 2 raise to minus l i, i is equal to 0 to K minus 1 is equal less than equal to summation pi, i equal to 0 to K minus 1 and this is equal to 1.

So, what it implies that if I if I choose this inequality which is satisfied by l i then I will get this result that 2 minus l i, i is equal to 0 to K minus 1 is less than equal to 1 and so if this condition is satisfied we know that we can generate a prefix code, correct.

(Refer Slide Time: 21:51)



So, it is reasonable for me to assume this choice of l i. So, if this choice is acceptable now, so if this is acceptable then I can see that if I have satisfied this constraint then multiplying by p i throughout and summing up or all is I will get it as, correct. So, this implies that my H s is less than equal to l bar is less than equal to H s plus 1. So, I have proven this inequality. So, the length average length will be away from the entropy of a discrete memory source by just 1 binit, ok.

Now, this relationship establishes the fact that the variable length codes that satisfy the prefix condition or efficient source codes for any discrete memoryless source for source symbols which are not equally probable, correct. Now, in the next class we will demonstrate an algorithm to constructs such codes, ok.

Thank you.