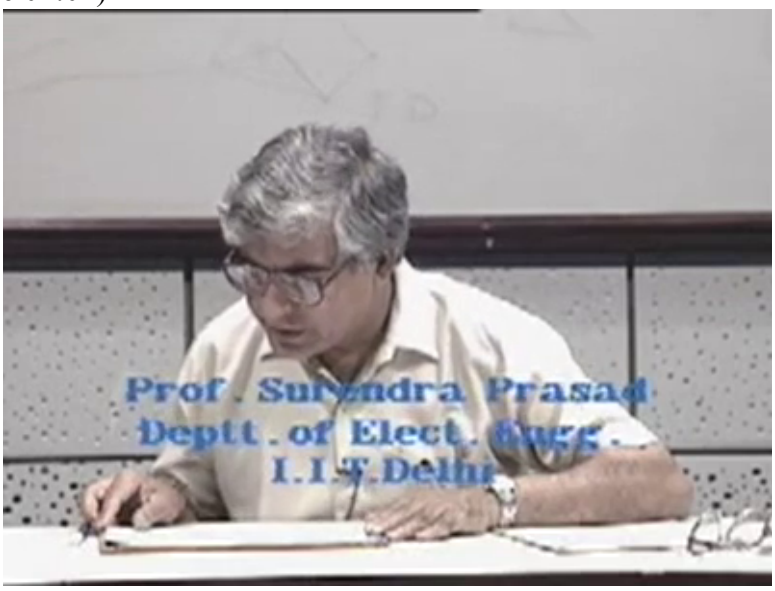
Digital Communication
Professor Surendra Prasad
Department of Electrical Engineering
Indian Institute of Technology Delhi
Lecture No 34
Source Coding

(Refer Slide Time 01:04)



We talked about the concept of information and how it is to be measured or defined, both from intuitive as well as engineering point of view. And we also looked into the definition of what we call entropy of a source in terms of both the information measure as well as entropy of the source are attributes of the source itself. And they are really defined by the message probabilities, this source symbol probabilities associated with the source.

(Professor – student conversation starts)

Student: Sir, 0:01:40.3 the continuous distribution theorem?

Professor: As I said, for the time being, for convenience I am taking discrete sources. If we have time we will look into continuous sources. I plan to do that, right? But for the time being we are looking into discrete sources because it is easier to formulate these concepts in terms of discrete sources. The basic definitions will remain the same, they will not change from when you from discrete to continuous. The concepts will be the same but some of the details will obviously be different.

(Professor – student conversation ends)

In any case we also looked into the concept of entropy and by taking an example of equi-probable sources that is, a source in which the message probabilities are all equal, we could appreciate the fact that it is possible to represent this source by an average number of bits which is given by the entropy of the source, average number of bits per symbol. I am representing each symbol, on an average, in fact, for equi-probable, all symbols will require same number of bits, right? For equi-probable sources all symbols will require the same number of bits and that will be equal to the entropy of the source.

And another result we discussed was that for entropy of the source to be maximum, all the message probabilities should be

(Professor – student conversation starts)
Student: equal
Professor: equal. So these are some of the results that we discussed last time.
(Professor – student conversation ends)

Now we will continue on that discussion today and try to show, to start with, that even for a general source in which the various message symbols may not be equi-probable, it is possible to encode it in such a way

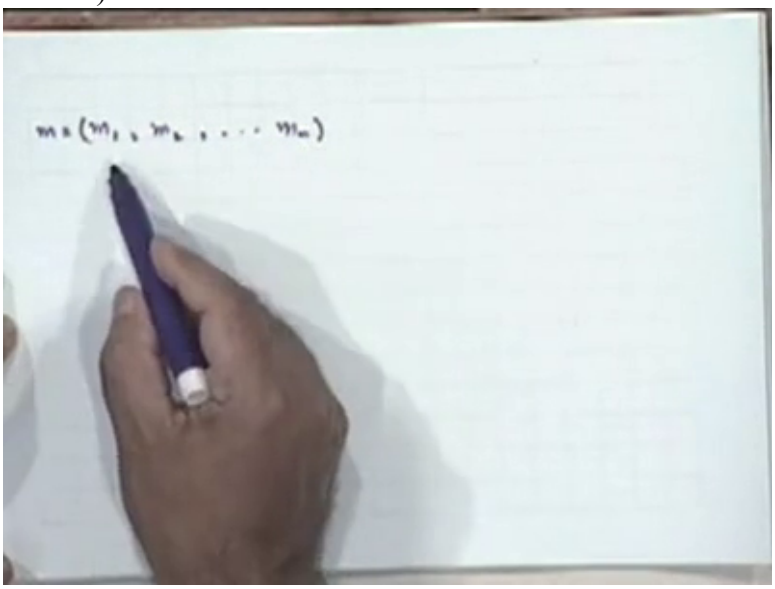
(Refer Slide Time 03:32)

that the average length of the code required to represent every symbol of the code, every symbol of the source is still equal to the entropy of the source. That is, somehow entropy is to

be largely now interpreted as a measure of what is a minimum number of bits required, or on an average how many bits are required to represent the source digitally using binary digits, Ok.

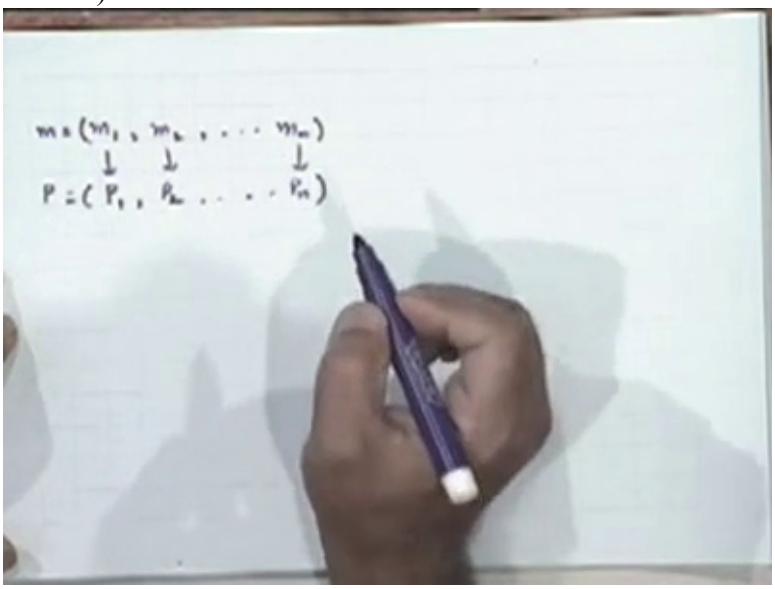
So to discuss that, as I said, I think we had already started on this aspect, let us consider a source m with message symbols given by, n message symbols

(Refer Slide Time 04:17)



m sub 1 to m sub n. Let us say this is associated with the probability vector p sub 1 to p sub n. That is message symbol m 1 is associated with the probability p 1, m 2 with p 2 and m sub n with p sub n.

(Refer Slide Time 04:37)

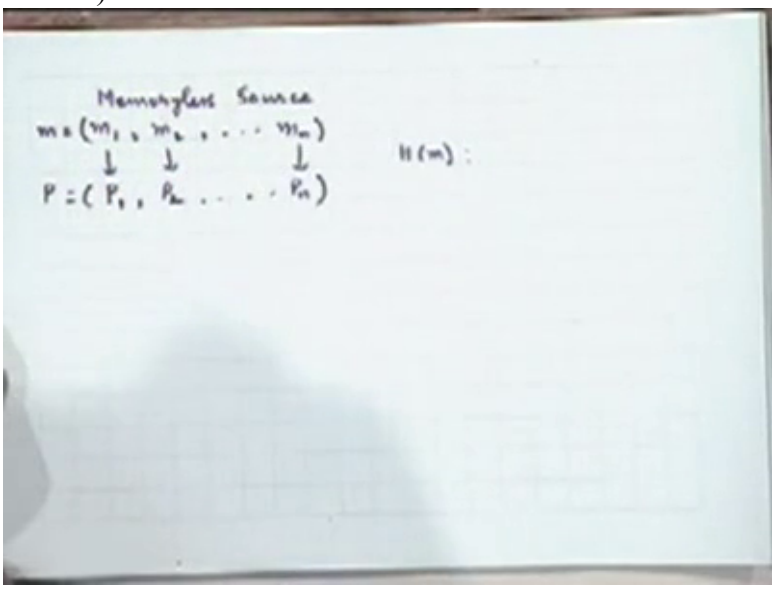


What I want to demonstrate, that even for a source distribution of this kind, it is possible to think of entropy as the average number of binary digits represent to, required to represent each symbol of the source; the average value of course. The actual value will be different from symbol to symbol, right.

And I will try to demonstrate this through construction that is by showing a procedure, an encoding procedure wherein this will be possible, Ok. So let us look at that. The construction procedure that we will adopt is as follows.

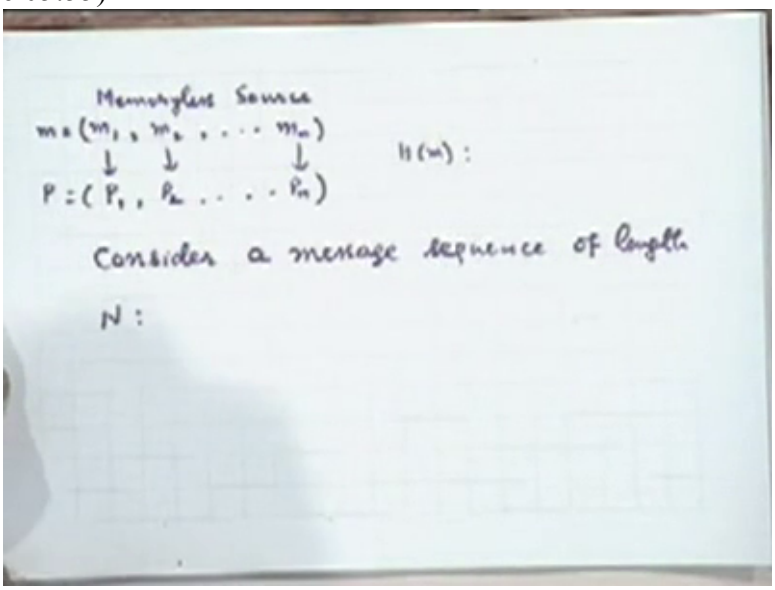
As I said, we are considering a memoryless source. This is another assumption that we are making for the time being, alright?

(Refer Slide Time 05:28)



What is a memoryless source? That is from symbol to symbol the emissions are independent, uncorrelated. So we will consider a message sequence coming out of this source, right? Let us say the length of this sequence; the message sequence is n, Ok.

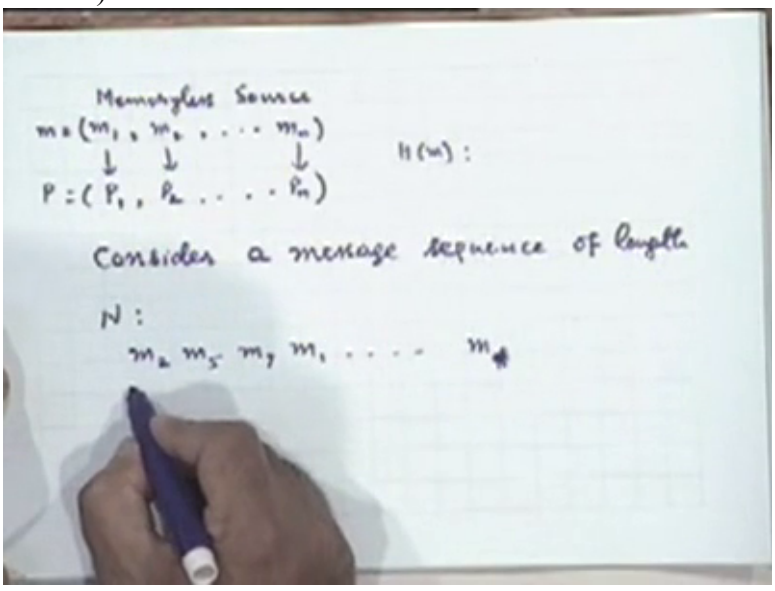
(Refer Slide Time 05:55)



So in the first, the first entry of the sequence will be one of these symbols, the next one will also be one of these symbols but picked up randomly and independently from the previous one and so on and so forth.

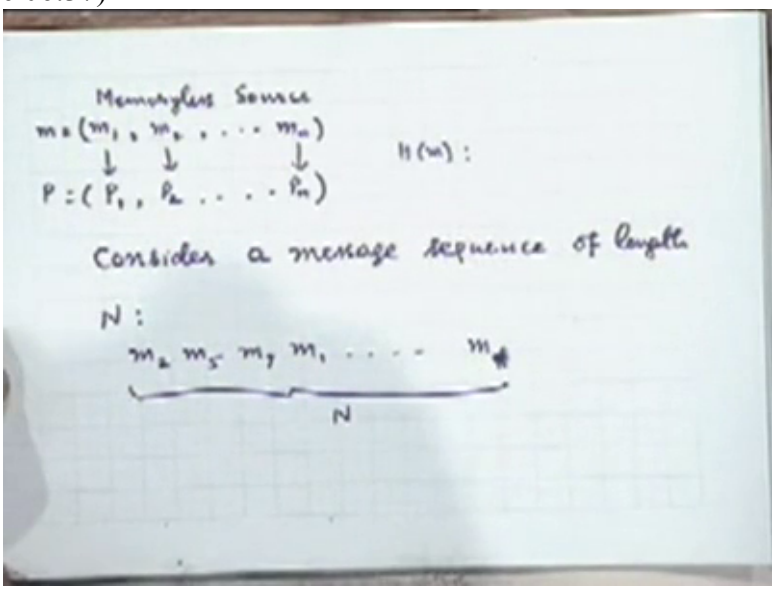
So we will essentially have, let us say some arbitrary message sequence may be, this may be a typical example, right? Gong up to, going up to some value whatever it may be. m sub k where k goes from 1 to n. But the

(Refer Slide Time 06:32)



total length of the sequence we are considering is N,

(Refer Slide Time 06:37)



arbitrary, where N is arbitrary, N is arbitrarily large to start with, Ok. Picked up randomly from this message because let us say these are the conse/consecutive, let us assume these are the consecutive emissions from this source in every succeeding interval of time, whatever the unit of time, fine?

So let us just analyze the structure of this N-symbol message sequence that we have got to start with, right? Let us try to analyze it for how many of these symbols are going to be m sub 1, how many of these are going to be m sub 2, and so on and so forth. What can we say about that? Provided that N is sufficiently large, right, that is, if N tends to infinity, it is quite easy to argue from the basic understanding of probability, definition of probability that you might have as essentially relative frequency of occurrence of a particular event, that is how you understand probability, relative frequency of occurrence of a event in relation to the total number of trials, right.

So if our total number of trials is N, that is total number of symbols I am considering is N, N is very large, N tending to infinity, what can we say how many times the symbol m 1 will appear in this message sequence?

(Professor – student conversation starts)

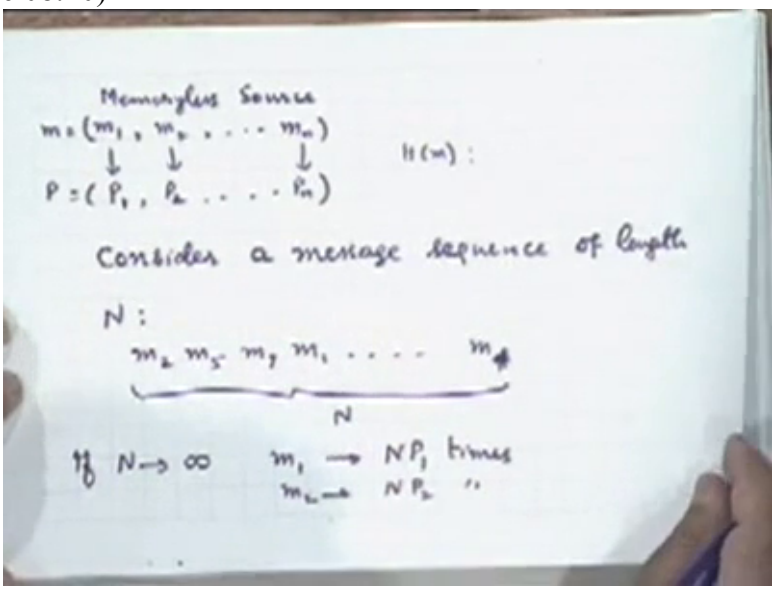
Student: p 1

Professor: p 1 times

Student: N

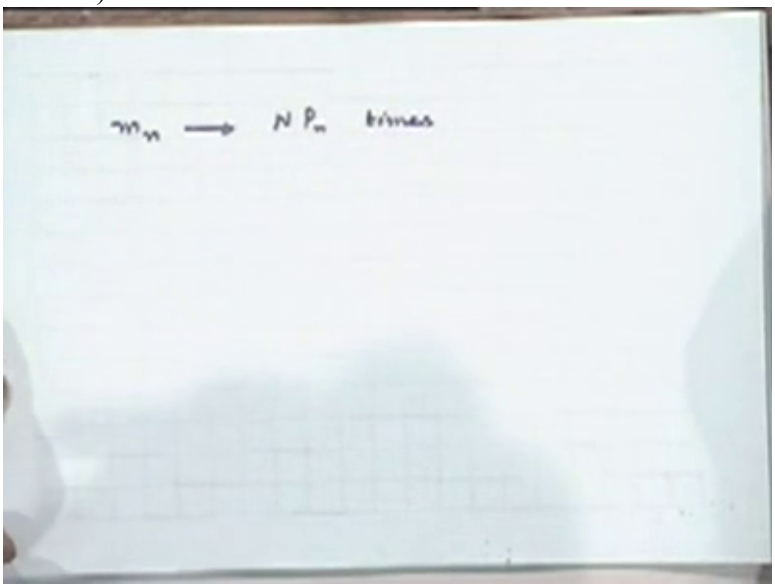
Professor: So m 1 will appear N times p 1 times. Similarly m 2 will appear N times p 2 times and so on.

(Refer Slide Time 08:20)



m sub n will appear n times p n times,

(Refer Slide Time 08:31)

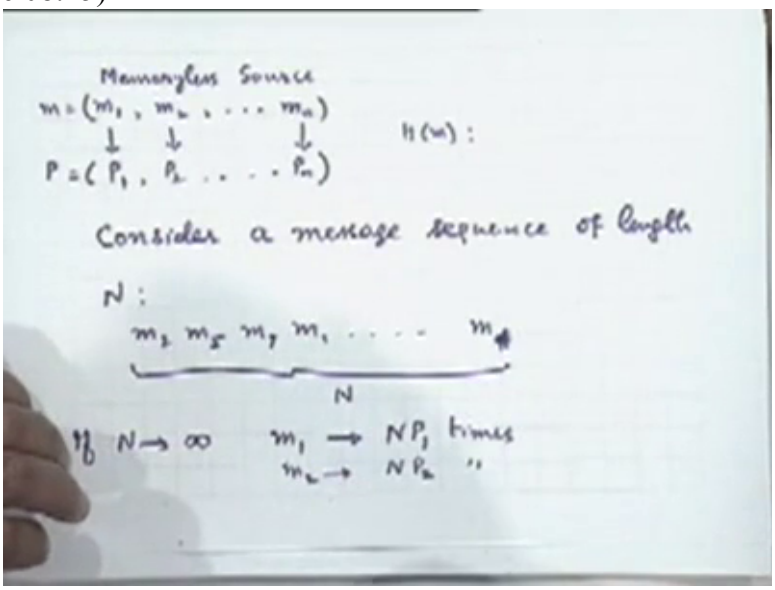


Ok. And this fact is independent of what particular message sequence is actually being considered, right?

(Professor – student conversation ends)

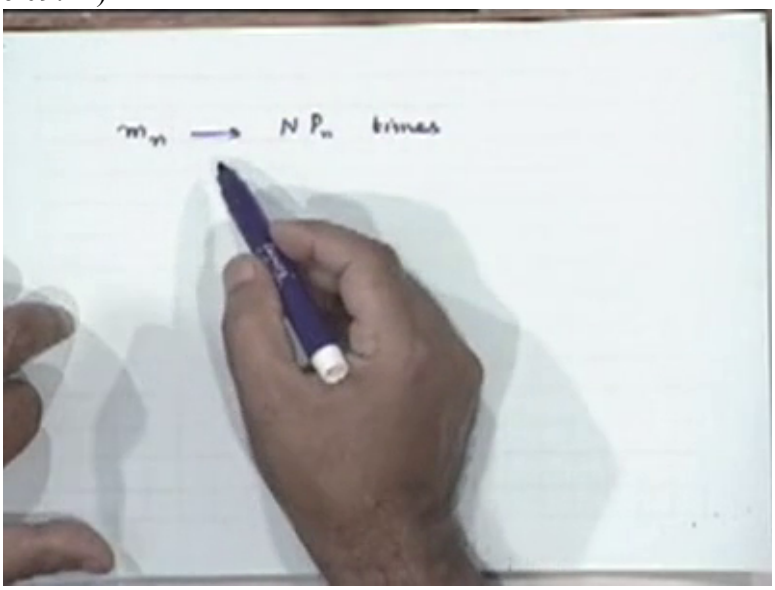
This message sequence may be different from one instance to another instance,

(Refer Slide Time 08:45)



right? In this case I have got this particular sequence. In another case I might start with m 1, go to m 3, m 5 and so on and so forth, right? No matter what particular message sequence of length N is being considered the structure, the structure of the sequence will be such that it will have so many times, so many occurrences of m 1, N p 1 times, so many occurrences of m 2 and so many occurrences of m sub n,

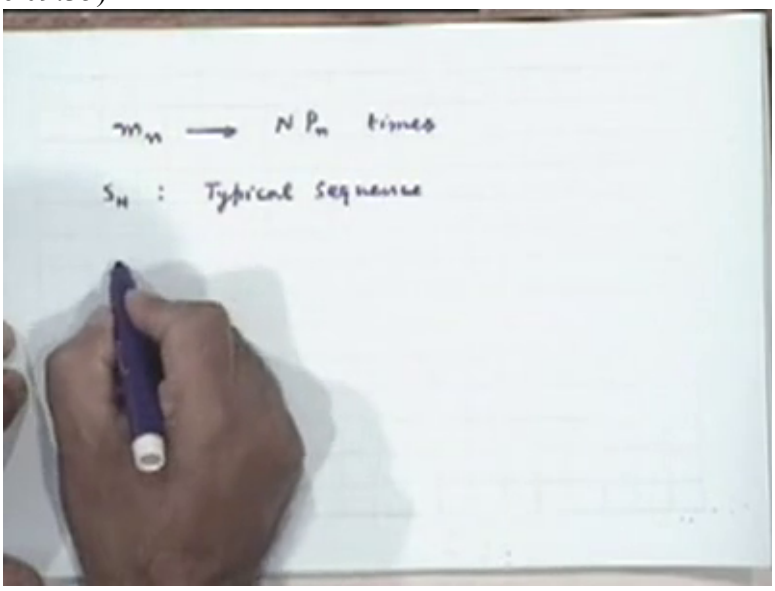
(Refer Slide Time 09:14)



alright. This is independent of the actual sequence being emitted by the source in a particular situation.

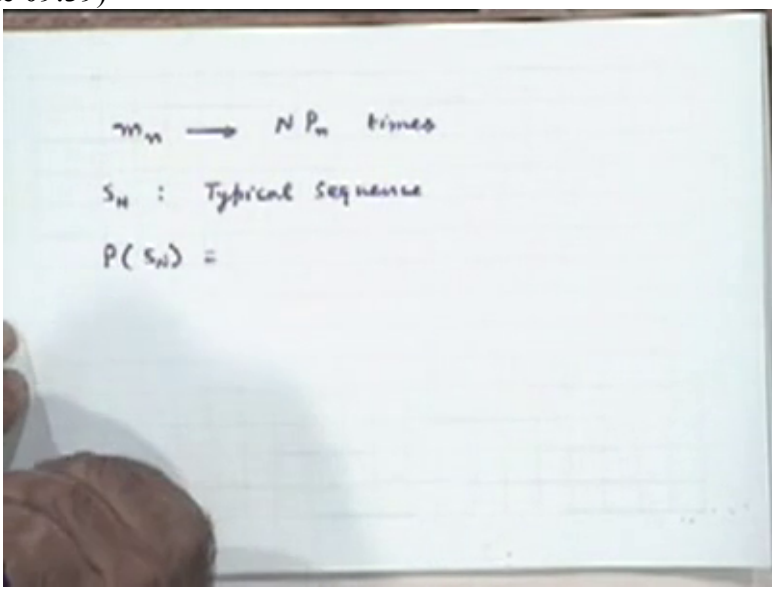
So if I call this typical sequence S sub N, right, this is just a notation for the typical sequence that you are just considering.

(Refer Slide Time 09:35)



Then we can say that the probability of the typical sequence S sub N,

(Refer Slide Time 09:39)



what is this going to be equal to?

(Professor – student conversation starts)

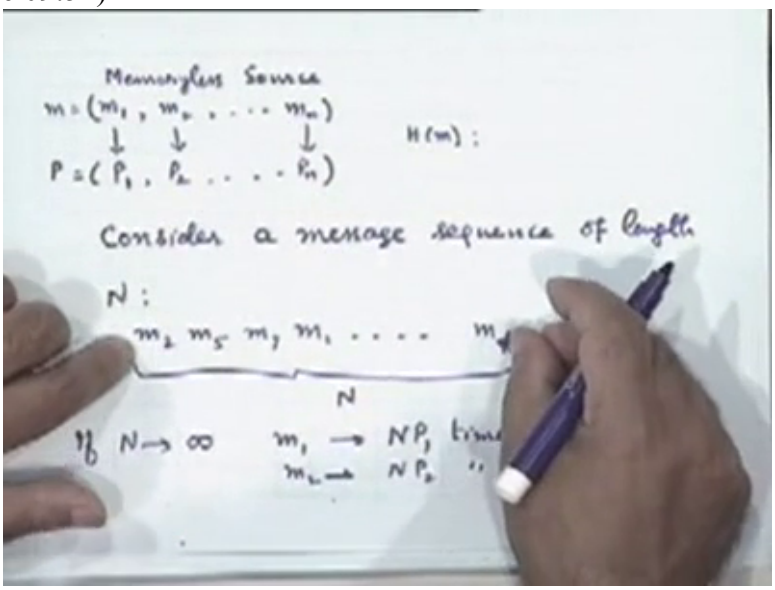
Student: Probability of the...

Professor: Probability of occurrence of this specific typical sequence we are considering?

Student: an independent

Professor: Let us say, what is the probability that this specific sequence

(Refer Slide Time 09:52)



would have occurred?

Student: Sir, order of the...

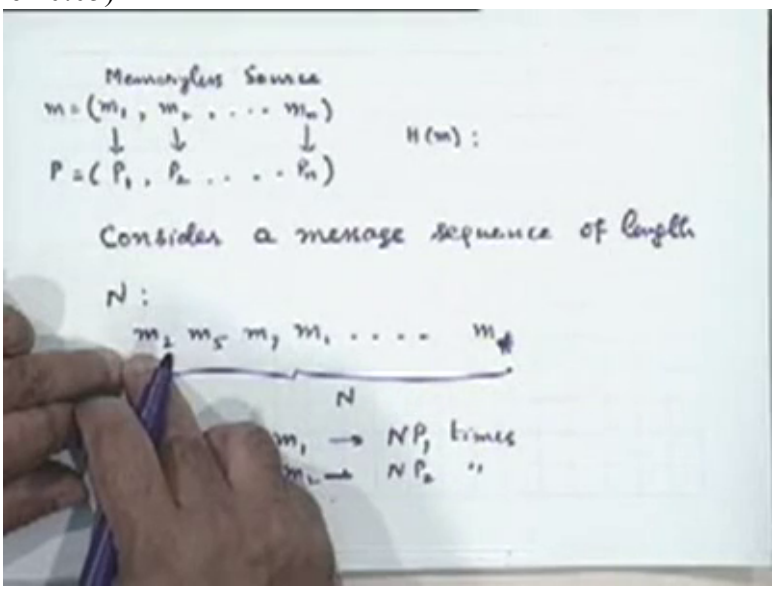
Student: Order of

Student: Multiplication of these

(Refer Slide Time 09:56)

Professor: Well, will order matter? Order will not really matter. What is going to really matter is

(Refer Slide Time 10:05)



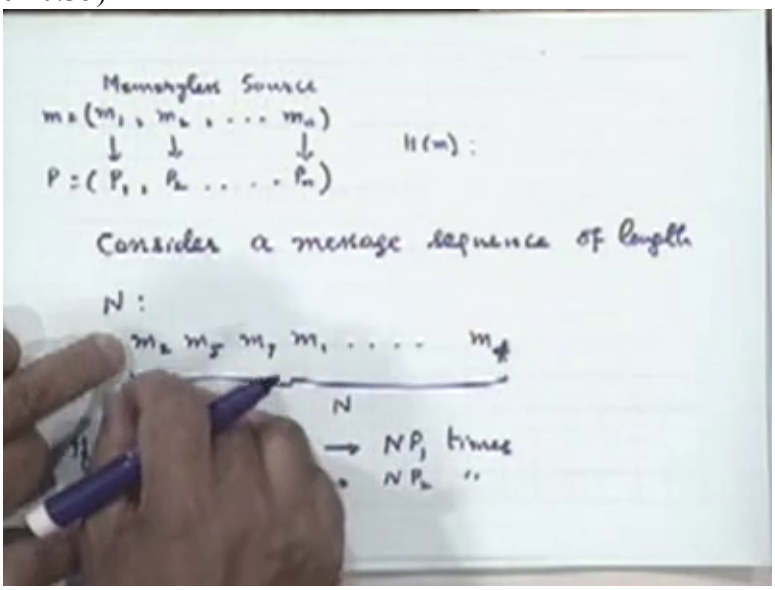
Student: Number of message sequences possible, 1 by N

Professor: Yeah best thing is to look at the total

(Refer Slide Time 10:13)

number of sequences possible and then 1 by N of that is, that is one way of looking at it. Alternatively we could, we could just appreciate that each of these symbols is being independently be emitted, alright. So the specific sequence that this has been, will be obtained will be simply

(Refer Slide Time 10:35)

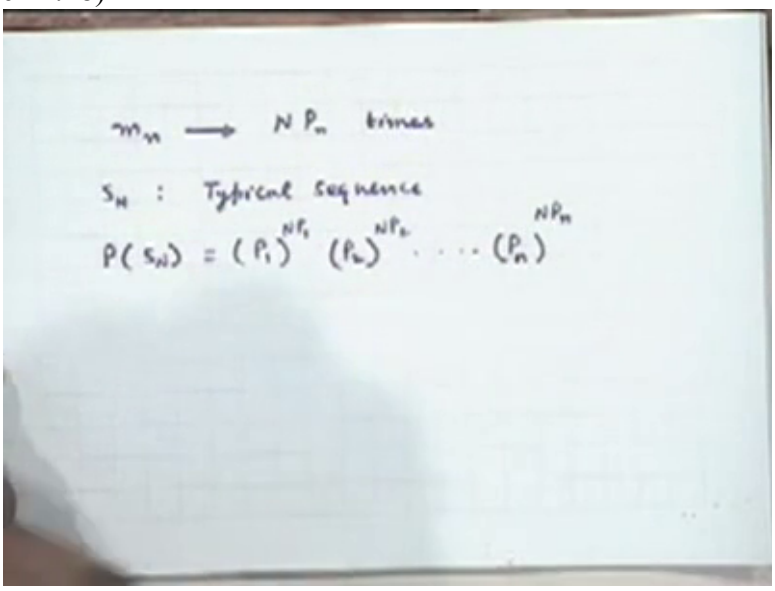


the product of the probability that m 2 occurs here, m 5 occurs here and so on and so forth, right, because of independent emission of sequences. So probability of a specific sequence S sub n is going to be p 1 to the power

Student: N p 1

Professor: N p 1 because m 1 will occur somewhere along the line N p 1 times, right? Similarly p 2 to the power N p 2 and p sub n, small n to the power N p sub n.

(Refer Slide Time 11:18)



Do you all agree with this? Amitabh, some questions there? Tarun?

(Refer Slide Time 11:32)

Student: Shouldn't p be equal to 1 by N because...

Professor: 1 by N? No

Student: N

Student: Equally probable, this p 1, p 2

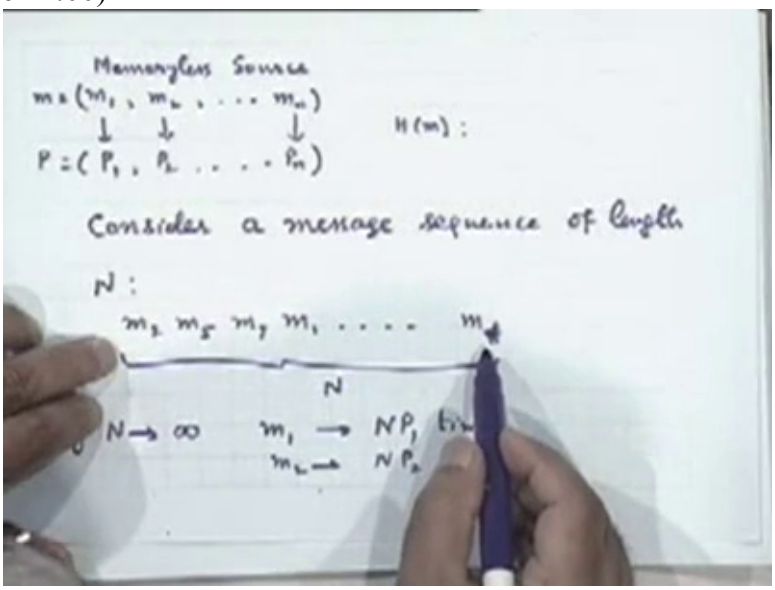
Student: Otherwise how can this whole sequence be possible? 1, equal to probabilities are only 1 by N, isn't it?

Professor: N sequences are possible, what does that mean?

Student: k sequences are possible.

Professor: Right, one could take that approach and one will get to the same result, right? This is easier to appreciate, right? That is your sequence consists of a number of independently emitted symbols,

(Refer Slide Time 12:00)

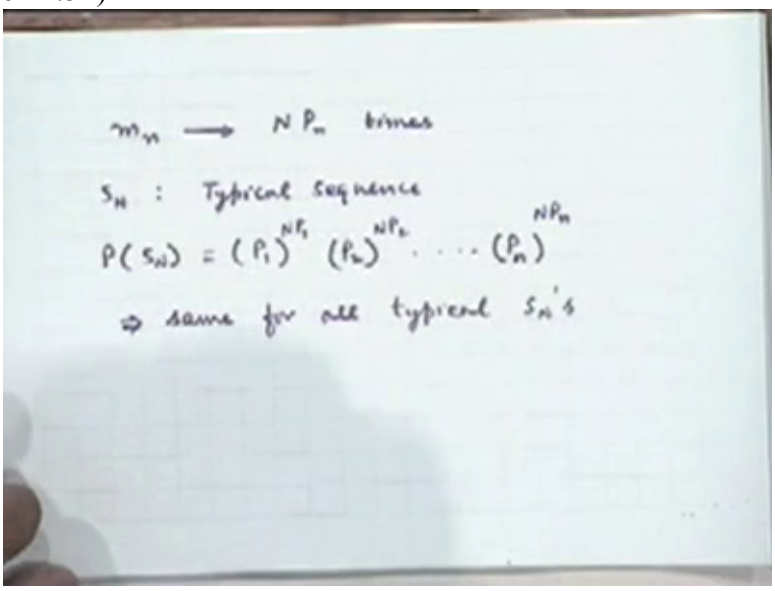


right? Therefore the probability of this specific sequence will be the product of these events, the product or probabilities of these events which is what you are writing. This is the simplest thing to mathematically write.

(Professor – student conversation ends)

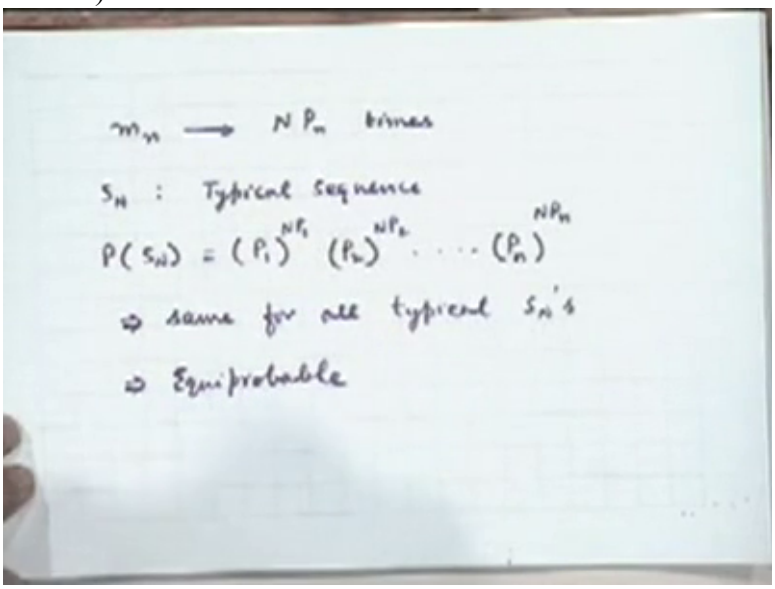
And the next important thing to appreciate is that this probability of a typical sequence that we have considered will be the same for all typical sequences, will be the same for all sequences for that matter because all sequences have the same precise structure in terms of m 1, m 2, m sub n, right, provided N is sufficiently large. So it is same for all typical S ns.

(Refer Slide Time 12:52)



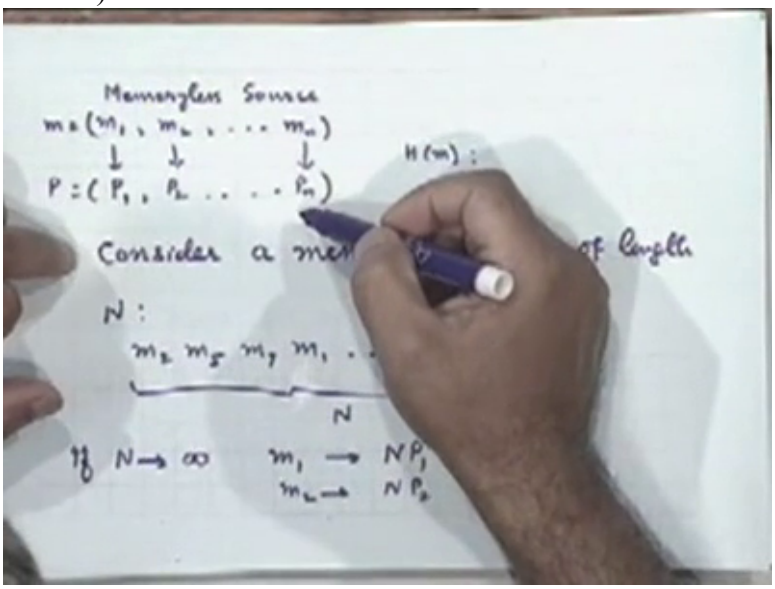
What does it mean? That this sequence of this n, n long sequences, that message sequences that we have constructed provided N is sufficiently long, sufficiently large are all equi-probable,

(Refer Slide Time 13:11)



Ok. So this is the trick that I am playing. I have constructed from the original message source which was not equi-probable,

(Refer Slide Time 13:22)



a new message source which is equi-probable, right where the new message source emits not a single symbol but a sequence of N symbols where N is very large, Ok. Is it clear? And I have got an equi-probable source.

(Professor – student conversation starts)

Student: And in each sequence the number of, the number of times which the m i

(Refer Slide Time 13:48)

repeats is fixed.

Professor: Yes, it is fixed in terms of these probabilities, right? So I have got a new source which is equi-probable, right?

Student: Sir is that 0:13:58.4 fixed value, number of times...

Professor: Well if N is sufficiently large it is fixed. N p 1 times will be fixed number, right as N tends to infinity. It will be some constant value.

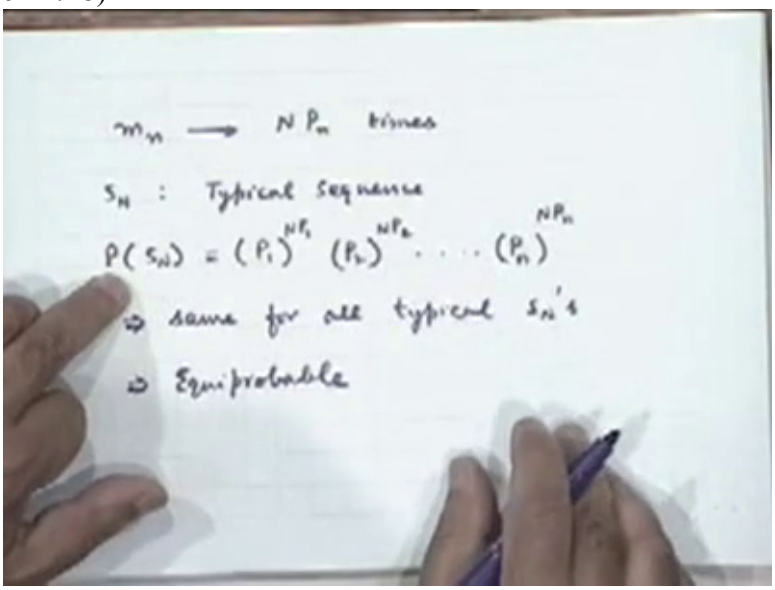
Student: That means you can still differentiate between 2 sequences.

Professor: Yes, of course. The sequences are all different, right?

Student: Probability is the same.

Professor: But they all have the same probability. And the probabilities are each equal to p, this probability.

(Refer Slide Time 14:28)

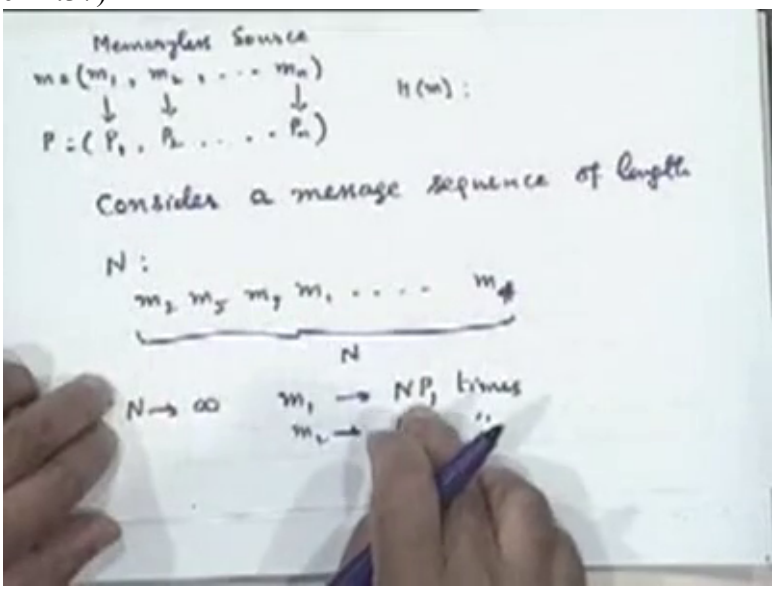


Student: The sequences are same.

(Refer Slide Time 14:31)

Professor: No, the sequences have the same structure, structure in the sense that m 1 occurs so many times

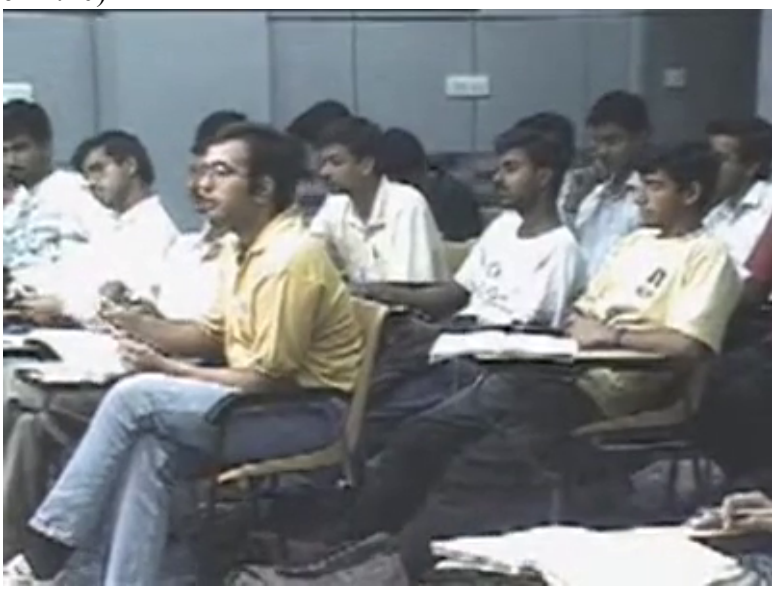
(Refer Slide Time 14:37)



but where do they occur?

Student: The permutation is not the same.

(Refer Slide Time 14:40)



Professor: The permutations are different. Where do the precise locations of m 1, where are the precise locations of m 1? They will be different from sequence to sequence, right. But they all have the same number of occurrences of m 1, same number of occurrences of m 2, and so on and so forth. So the sequences are really different. But their probabilities are the same. And that is finally reflected in the fact that they all have the same probability of occurrence, Ok.

Student: But Sir

Professor: Yes

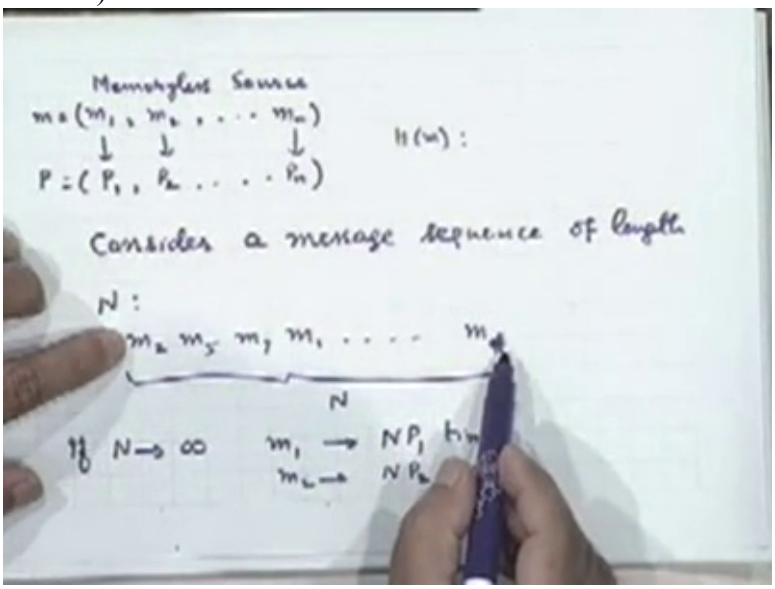
Student: This is not going around; first we are finding the number of times a particular symbol occurs from assuming the size. Then we go back to the probability, I mean.

Professor: What is wrong with that? I do not understand your...

Student: Find the number from the entire sequence and then find the probability of indirect sequence...

Professor: Yeah, one could do that and that would be a slightly more lengthy procedure but what is wrong with this argument? I want; you said that there is something wrong. I would like to understand what is wrong here? Because my argument is that you have a specific sequence,

(Refer Slide Time 15:43)



right in which these various message symbols are emitted independently, each with the probability as given over here. Now if I want to find out the probability of this specific sequence well I have to find out that m 1 occurs here and that occurs with a probability p 1, it will also occur somewhere else with the same probability, somewhere else with the same probability, total of N p 1 times. What is the probability of these happening? p 1 to the power N p 1, right?

Student: Sir, p 1 is independent no, how will it form sequence?

Professor: Sorry?

Student: These are independent.

Professor: Yes

Student: How can they form sequence?

Professor: Ok, I think may be something is wrong in my presentation. Let us quickly go through what we are doing.

(Professor – student conversation ends)

We have to start with a message source whose dictionary is this n message symbols. Now we are thinking of this message source emitting a sequence of symbols from its dictionary one after another. And we are looking at N such emissions, right? That is we are not trying to encode each emission as it is emitted but we are letting them accumulate over a length of N symbols and then we are thinking of coding them, right? That is, the reason why I am doing that is because I want to construct an equi-probable message source out of this non equi-probable message source.

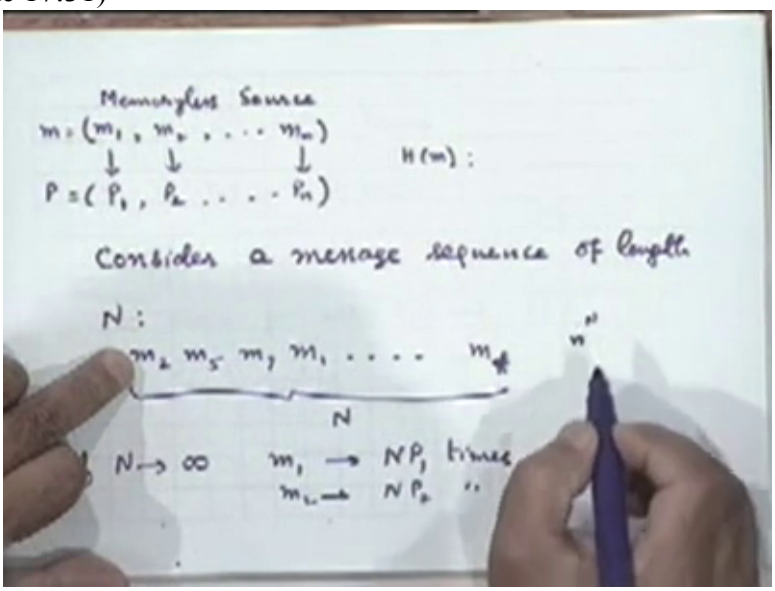
This new source that I have constructed, its dictionary is all possible sequences of length N of this kind, all possible sequences, right? What will be the value of such sequences? What will be the number of such sequences?

(Professor – student conversation starts)

Student: n raised to N

Professor: n to the power N, small n to the power capital N,

(Refer Slide Time 17:31)



right?

Student: Then Sir, the probability of getting the sequence will be equal to 1 divided by n raised to N.

Professor: This is precisely it will turn out to be, right.

Student: We have considered the case of number of, the number of times each m i appears is fixed. So in this case, this won't be valid.

R discussing

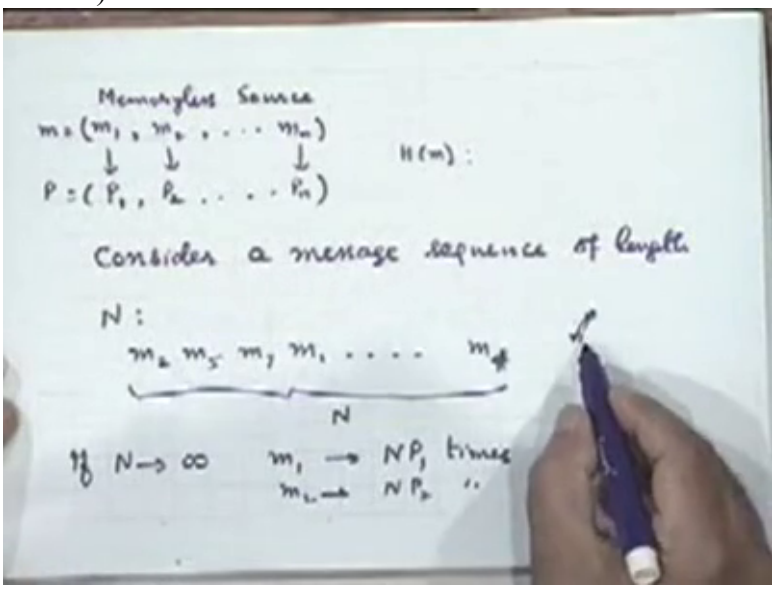
Student: So in each S of any random n, the number of times which m 1 appears is fixed. In that case the probability cannot be equal to 0:18:05.8

R discussing

Student: This is like...

Professor: Yeah, let me just... maybe this statement is not correct then. Yeah, that statement will be true

(Refer Slide Time 18:20)



if each of these were already equi-probable. I think you are right. I should not make that statement, yeah.

Student: What sir?

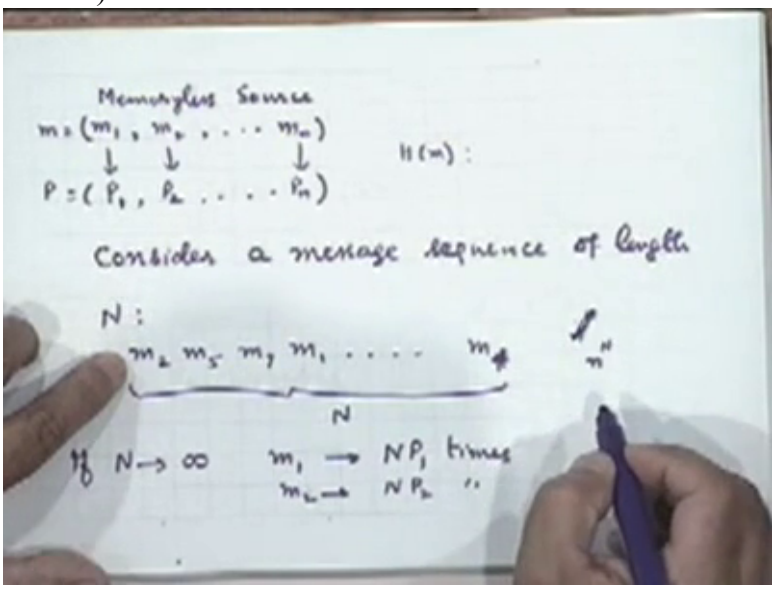
Professor: That is, if each of these were equi-probable, then perhaps I could have made that statement.

Student: Which statement?

Professor: That is, the total number of possible kinds of sequences we can have here is small n to the power capital N.

Student: No, no

(Refer Slide Time 18:44)



Student: Capital N to the power capital N.

Professor: How can it be capital N to the power capital N?

Student: The N.... will be only N then...

Professor: Because, because we cannot have, we cannot have an arbitrary number of m 1 or m 2 or m 3 in this sequence.

Student: Sir, this small n to the power capital N would be valid when each of the symbol is uncorrelated with the previous one.

Professor: That it is.

Student: But even otherwise assuming that although they are uncorrelated, if we take a 0:19:16.3 the number of...

Professor: We are not assuming that. That comes from the definition of probability. That comes from what the relative frequency implication of probability. This is not an assumption.

Student: Sir if we are assuming that...

Professor: No, we are not assuming that, Ok. If we have N very large, Varun, please let there not be so many parallel discussions. If there is a confusion let us try to resolve it here. If we have N trials, iIn each trial we have, these are the options available to us with these probabilities. This is the experiment we are conducting.

Student: Sir, the point is this. When we talk of probability we talk about some correlation, don't we?

Professor: That correlation is zero. The correlation from one emission to the next emission is precisely zero. We are saying they are independent. Now this occurs with a probability p 1.

This also occurs independently what has occurred here. This is the correlation you are referring to? Ok

Student: Sir, especially one thing will lead to another.

Student: No, it is Ok, sir. Two states are independent, no.

Student: Sir these are independent and number of m 1 and m 2 are fixed. Then there will be only one sequence.

Professor: No

Student: Sir, how will you arrange this...?

Professor: It is clear to me that your basic fundamentals of probability are not very clear.

Student: Sir how do we arrive at the 0:20:48.4

Student: Sir, let us leave it Sir...Let us proceed further, Sir.

Professor: Ok

R discussing

Professor: Let us, I think we are going into circles without really achieving anything. So we will skip this discussion any further.

(Professor – student conversation ends)

In a nutshell, please think about it, and it is very easy to appreciate this fact that if your basic fundamentals of probability are understood that this fact comes from the definition of probability assuming that these n pickings from this source are independent of each other. That is all. And once this structure is there, I think it is quite obvious that specific sequence, a typical sequence will have this probability and this probability will be independent of what typical sequence you pick up, right? These are the broad points to understand.

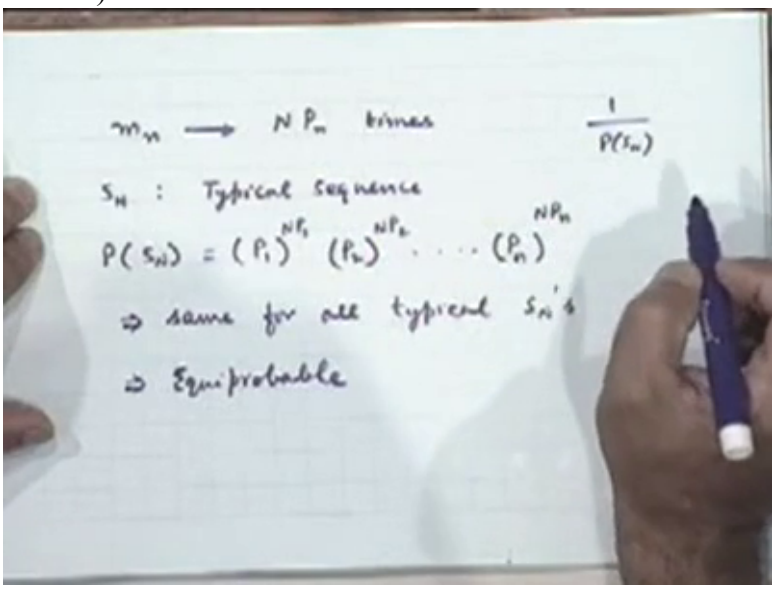
Don't try to confuse yourself by looking into, in a manner which just unnecessarily confuses you. These are the essential facts. The facts are the emissions are independent with these probabilities. We are considering N emissions, independent emissions. Therefore this leads to this structure of, this frequency of occurrences of m 1 to m sub n, right? Because of that frequency of occurrence of individual m sub is, a typical sequence will have this probability of occurrence. This was a broad fact.

(Professor – student conversation starts)

Student: Therefore the number of sequences would be equal to 1 by p S n.

Professor: The number of sequences would be equal to 1 by p S n, quite right.

(Refer Slide Time 22:20)



Student: How?

Professor: Because they are equi-probable. You are quite right. I am going to use that fact now.

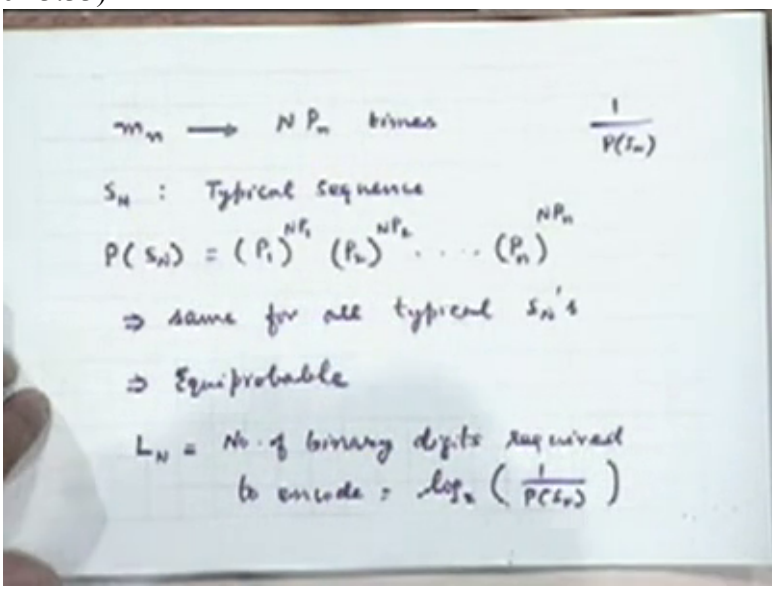
Student: All are equally probable and exclusive.

Professor: Ok? Alright. If you still have some doubts, let us discuss them separately later because I have spent too much time on this, much more than I had planned to.

(Professor – student conversation ends)

Now how many bits, what kind of, how many binary digits are required to encode this equi-probable message source, this I know now, right? What is the value? What is the, let me denote by L sub n as the number of binary digits required to encode this equi-probable source. It is equal to log to the base 2 of 1 by, of this probability, right, which is P S n,

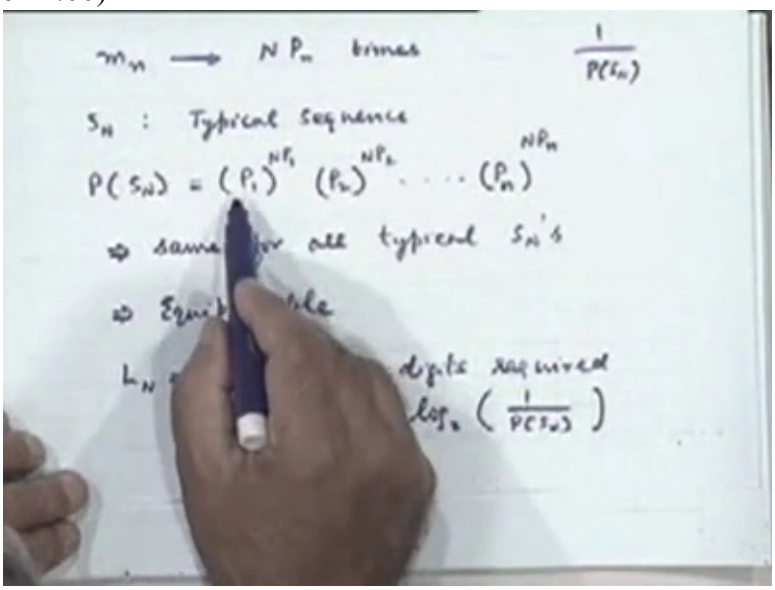
(Refer Slide Time 23:33)



right? That basically comes from this interpretation that you are just mentioning that we can have total of this many sequences of this length, right, of this structure because we are specifying the structure, Ok.

So let us try to simplify that. Substitute for p of S n

(Refer Slide Time 24:00)



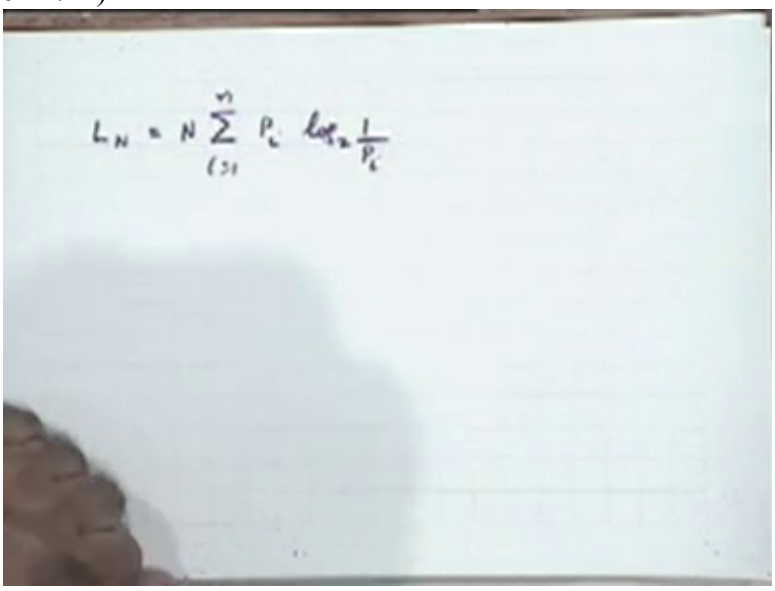
as this product of these probabilities, what will I get? This is 1 by p 1 to the power N p 1, 1 by p 2 to the power N p 2 and so on. So if I take the log of that what will I get? Substitute from this expression into this and use the properties of the logarithmic function and that will give you n times p i log to the base 2 of 1 by p i, i going from

(Professor – student conversation starts)

Student: 1 to n

Professor: 1 to n.

(Refer Slide Time 24:42)

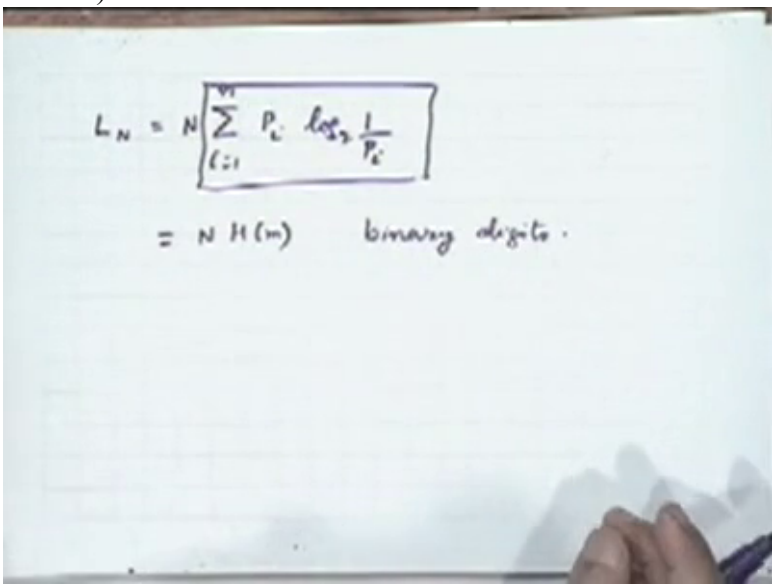


Is it clear?

(Professor – student conversation ends)

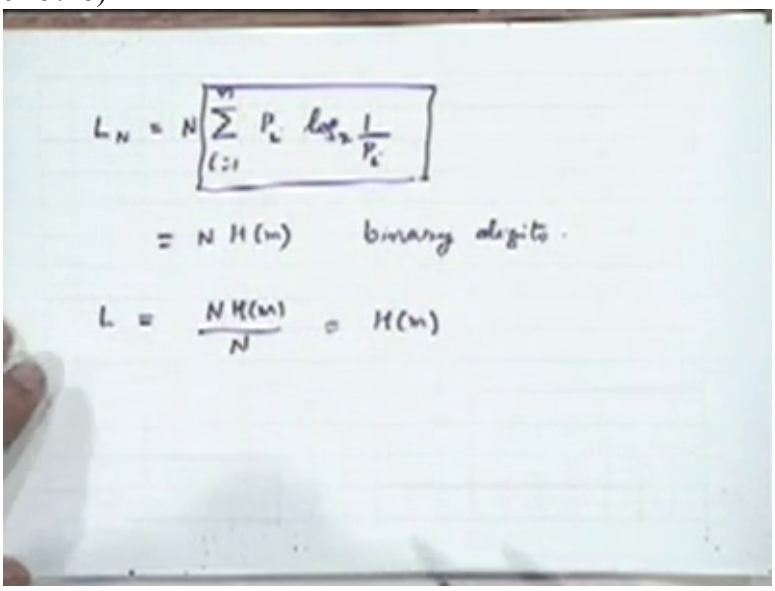
But what is the summation? That, by definition, is entropy of the source, Ok. So what do I find?

(Refer Slide Time 25:06)



That to encode a sequence or symbol coming from this new equi-probable source which consists of N original symbols, right, this new equi-probable source emits N symbols at a time of the original source; you require so many binary digits on an average, right? Therefore in terms of the original binary source, in terms of the original source what is the average number, what is the average length of the code? This divided by, because this represents N symbols of the original source, is equal to H which proves, what I wanted to show

(Refer Slide Time 25:48)

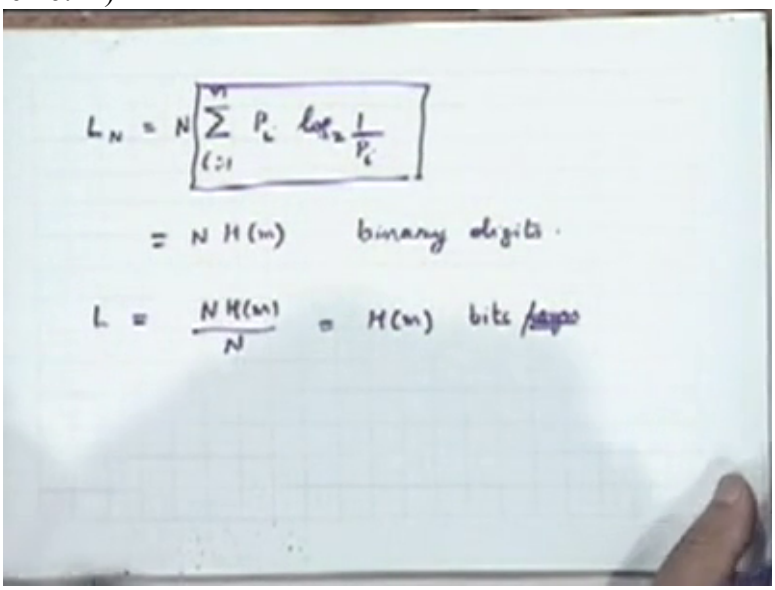


that one can, even for non equi-probable message sources have procedure by which I can represent on an average, each single emission by so many binary digits, so many bits because this is I am writing in terms of length.

(Professor – student conversation starts)

Student: Could you please show to the previous....

(Refer Slide Time 26:14)



Professor: Sure, what is your question? L is the length required to encode the individual symbol of the original source. Because this, this representation is representing N symbols, a sequence of N symbols.

Student: N is equal to for all the symbols, is it?

Professor: What is equal for all the symbols, L or? This is an average value so then average value is of course not an equal value, right? Average has a very specific meaning. I am assuming that you all understand what is average.

Student: Sir?

Professor: Yes

(Refer Slide Time 26:52)

Student: Sir, from this, 0:26:55.0 is it sort of data compression? Is this an...

Professor: This...

Student: Are you encoding sequence into 0:27:03.4?

Professor: It is not really, it is not like that. Yes, data compression is related to what I am discussing, right. But that is secondary because at the moment, there is no concept of compression coming into the picture. At the moment, I am asking the question, if I have a particular source with a particular entropy what is the minimum number of binary digits required to represent this source efficiently, right? And the answer is it is related to the entropy of the source.

(Professor – student conversation ends)

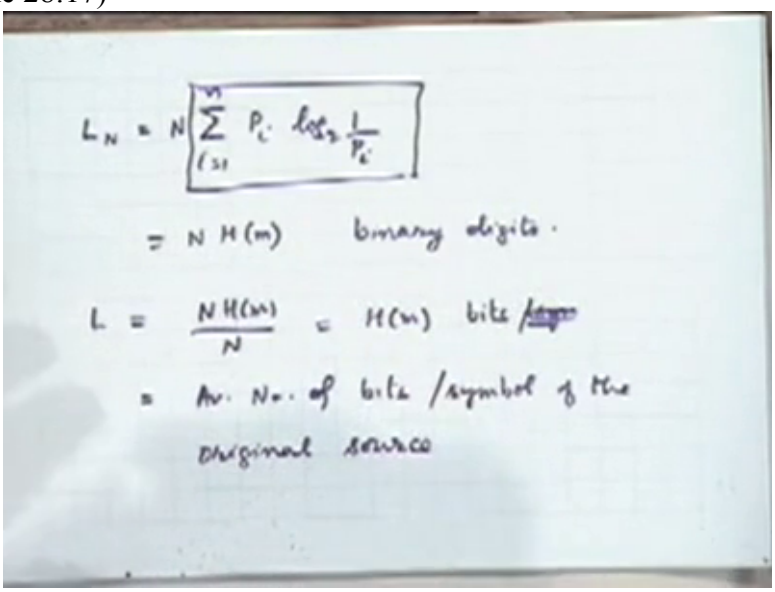
So when you are doing a data compression or when you are doing the data representation, you must bear this in mind, right? You will, your method of representing the source, how good it is, will be measured by this fact.

(Professor – student conversation starts)

Student: Sir the last...

Professor: Let me come to this point. In fact I am coming to; elaborate this point a little more. This is the average number of bits per symbol of the original source, right? Is it obvious?

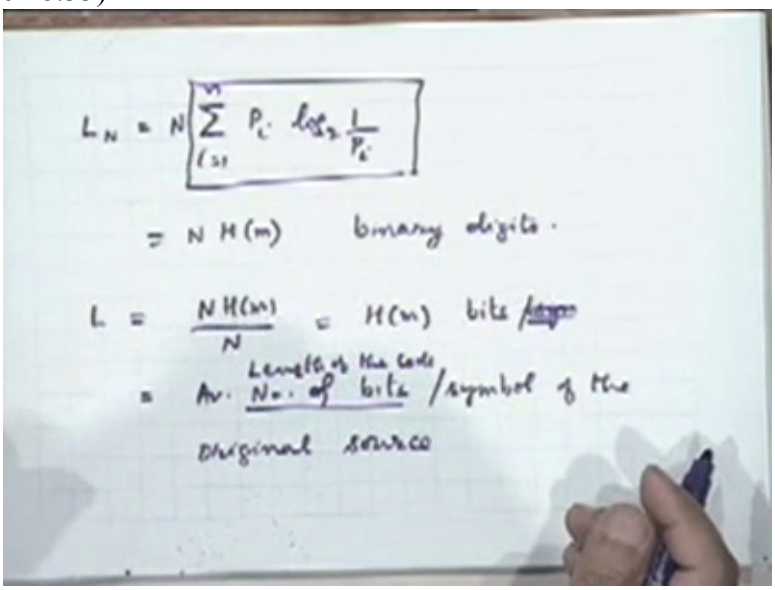
(Refer Slide Time 28:17)



Average number of bits, let us say average length of the code, per symbol.

Student: Capital L was the length of

(Refer Slide Time 28:33)



the total

(Refer Slide Time 28:34)

permutation or...

Professor: I had considered a group of N symbols which were being represented by so many final encoded sequence of this length, right? So per symbol basis on an average, you require simply H N bits per, of the original source, alright.

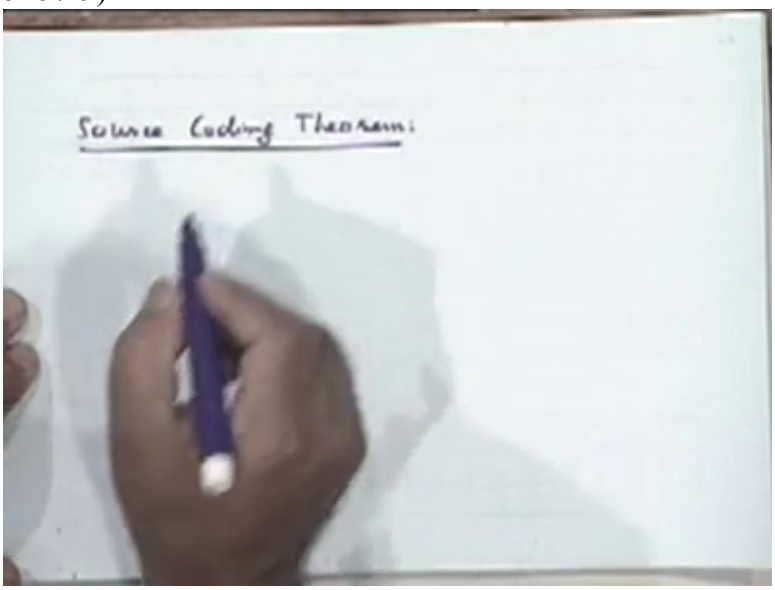
(Professor – student conversation ends)

Now it can be indeed shown, see what I have shown so far is that the average number of, the average code length required to encode this message is H m, the entropy of the symbol, the average over the symbols, the set of symbols that you are working with, right? It can further

be shown; I am not going to do that, the proof of that is rather tricky. In fact this is the minimum value that one can hope to get. The argument we have given is only for the average value, right? The minimum average value that we can hope to get is indeed the value of entropy of this source, right? And that is precisely what the source coding theorem states.
Let me just

(Professor – student conversation starts)
Student: Sir is it not obvious?
Professor: It is not obvious. Because what the...
Student: It is just the minimum...

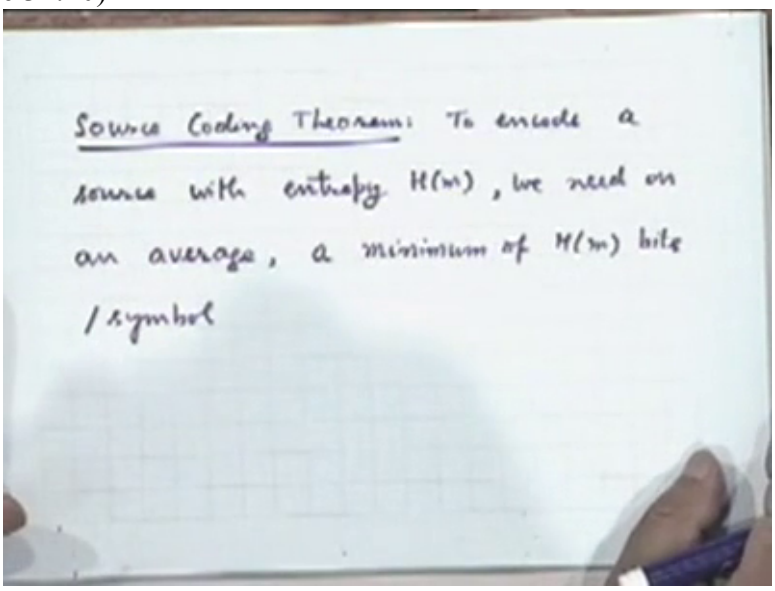
(Refer Slide Time 29:49)



Professor: How do we know that it is minimum? You have to prove that there is no coding procedure that exists which will require finally a code length less than the entropy, right, you have to prove that. I have not proved any such thing. All I have proved is from intuitive reasoning, that this is, I can device procedures by which I can achieve this average length, right? But maybe there is a coding procedure which exists, which can take us below this, right? One has to explicitly prove or disprove that fact. That is precisely source coding theorem, or what source coding theorem does for us but we are not going into those elaborate mathematical proofs.
(Professor – student conversation ends)

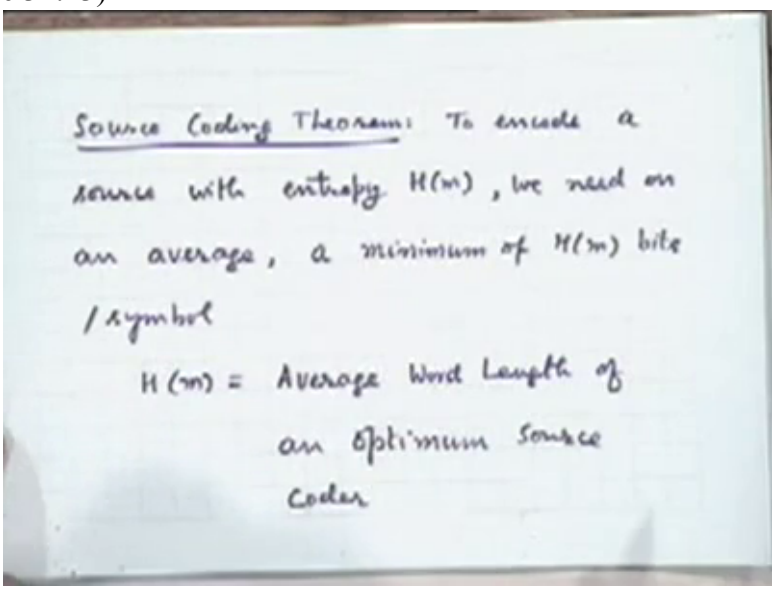
So the source coding theorem states that, to encode a source with entropy H m we need on an average, a minimum of H m bits per symbol or per message, Ok

(Refer Slide Time 31:10)



and you cannot do better than that. So in other words, we can expect your optimum source coder to produce an average word length equal to entropy,

(Refer Slide Time 31:43)



Ok.

(Professor – student conversation starts)

Student: Sir, what makes this on an average?

Professor: On an average, because your different symbols have different probabilities. Your different symbols therefore may be requiring different lengths for representation, right?

Student: Average length

Professor: But the average length

(Refer Slide Time 32:02)

is entropy, right? Average over the symbols.

Student: The minimum average

Professor: This is minimum value of the average that you can expect.

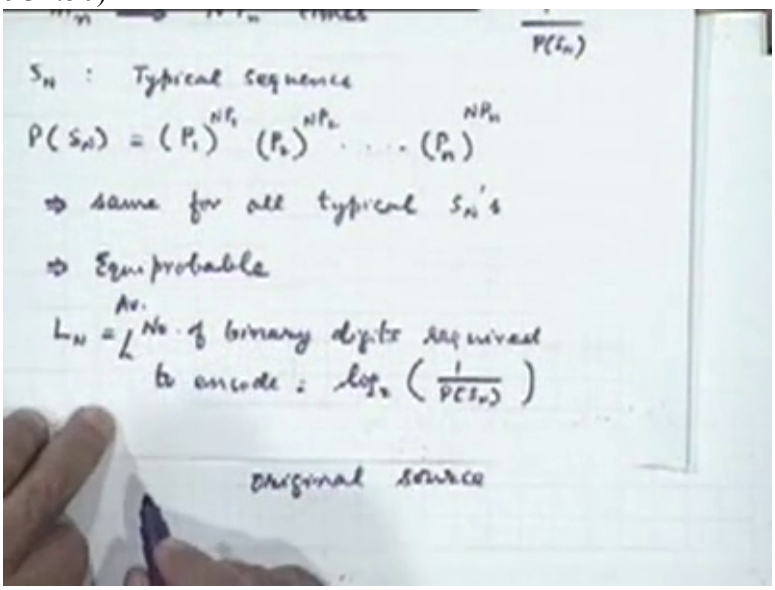
Student: It is coming out because you are actually taking the average of number of, total number of sequences possible and dividing by number of size.

Professor: You see that average is implicit because I consider equi-probable sources. For equi-probable the average is equal to the length of each symbol.

Student: Average is taken; it is not arithmetic but a probabilistic average.

Professor: It is a probabilistic average over the symbols because each symbol has a different probability, right? Is it clear? There also the average is being taken, when I wrote this expression, right, this was the average value. In fact I should write here average.

(Refer Slide Time 32:50)



It so turns out that in the equi-probable case, the average and the actual length are the same, right? So everywhere we are really dealing with average values.

(Professor – student conversation ends)

Ok, what is the implication of this? The implication is that, in order to efficiently utilize the channel on which you are doing your transmission, you know, one of the first things you should think of is, making sure that, you see ultimately we are going to have limits on how fast I can put data on the channel, right? And how fast I need to put data on the channel will depend on the kind of source emissions I have. How fast the source is emitting data for transmission on to the channel, right?

Now if I can encode by source representation properly, if I make sure that the representation is as efficient as theoretically possible, then I will be putting on the minimum possible demand on the channel in terms of data rate requirements, right? That is the basic significance of source coding and in this sense, you know the kind of source coders that we discussed, we have already discussed some source coders for information without perhaps your knowing that they are source coders.

No, source coders are basically devices which essentially give you digital representation of your sources. And that is the very first thing that we started discussing in this course, namely your p c m, your delta modulator and adaptive delta modulation and those kind of devices,

right? They are all source coders. They are doing digital representation of your signals, right? Now you should go back and see whether those are efficient source coders or not.

(Refer Slide Time 34:36)

We already discussed n fact to some extent how to make them efficient. For example we said for analog to digital conversion, we could use compenders to make them more efficient, right? Making use of the probabilistic structure of the, you know, values that you are going to get out of the source.

But even with all those modifications you will find that your conventional source coders, for typical sources for which you will be using them will be very, very inefficient. For example if you put that 8-ary converter to represent speech signals then you will find that you are ending up using an average codelength out of the 8-ary converter which is much larger, much, much larger than the entropy of the speech source, Ok

(Professor – student conversation starts)

Student: Are they being replaced?

Professor: Of course. That is a very major important thing for communication systems in communication theories, right?

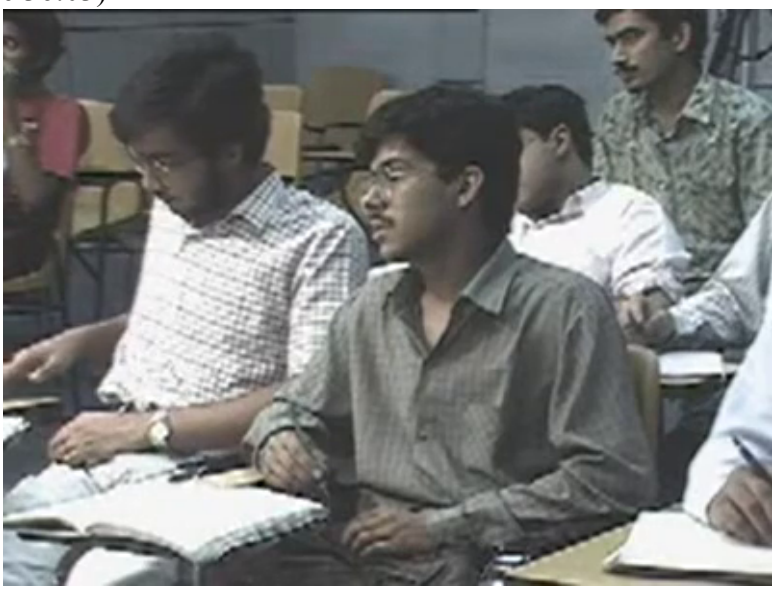
Student: How are these implemented?

Professor: We cannot answer everything in this question in this small little hour but we will certainly have occasion to discuss that. We have methods by which we can represent, you see now; therefore it depends on what kind of source you are representing.

(Professor – student conversation ends)

When you just put 8-ary converter, we do not really try to see what is the kind of signal that is coming into the 8-ary converter.

(Refer Slide Time 36:03)



Now we have to start seeing what is the kind of signal that is coming to the 8-ary converter. It is because what we are really trying to do now is represent efficiently that source, right, rather than just you will get straight-forward crude analog to digital conversion.

For example if it is a speech source, you have to understand the nature of speech signal, understand what is its entropy and then from these two understandings device method as to represent source, speech source by a rate as close to its entropy as possible, right? There are all kinds of techniques for doing that and they are being used in practical systems, Ok.

For example for speech we have formant vocoders, we have L P C vocoders, and there are all kinds of other source coders. Similarly for image, for images. The kind of compression, data compression techniques that are, these are called data compression techniques, right? Because basically we are trying to compress data over what you would do by 8-ary converter, straight forward you can't. Ok. So we will come back to this if we have time.

Now coming back to this discussion here, we have the method by which we can achieve the efficient source coding for discrete sources. Do you think it is a good method or a bad method

that we discussed just now? It is the best possible method? But yes in terms of achieving efficiency it is very good. But do we have some problems with this method? That is the question.

(Professor – student conversation starts)

Student: Provided we know the sequence, the...

Professor: No

Student: We have to know the probabilities

Professor: Yes, the probabilities you will have to know because the source has been modeled as the probabilistic source. So it has to be known, there is nothing, no disadvantage in that. One can study the probabilistic structure of you source.

Student: 0:37:49.9

Professor: No, a particular source, let us assume that a particular source has a structure. Now is there anything wrong? Let us put it this way. Let me make the question more specific. Suppose I have a probabilistic source in which I have N message symbols, each emitted with those probabilities p1 to p n.

Student: Sir, average, size of symbol is varying because....

Professor: That also is not a disadvantage.

Student: How do you, like...

Professor: In fact we expect, like in Morse code you know, we use different symbols for different, different codelengths for different symbols. It is in fact required. If you want to achieve entropy, that will be required. You cannot do anything about it.

Student: But actual, during transmission...

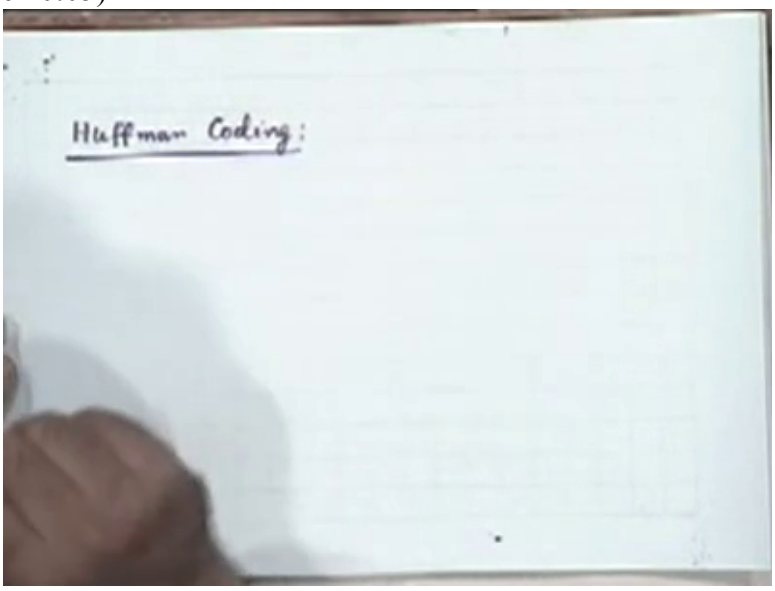
Professor: The real problem is you are not catching on to that, the value of N that we need is very large in this procedure, right? What does it mean? The coding delay is infinite. I have to wait for the source to emit infinite number of symbols before I can start encoding it. It is not practical, number 1. Number 2, it may be very annoying in real life situations. You may not, for example if you are trying to encoder for speech, and if your encoder itself requires you to wait for such a long time, the other guy will keep on waiting for your response after you have spoken, right? So it is not a practical way of doing things if N turns out to be very, very large.

(Professor – student conversation ends)

Therefore we would like to think of other coding procedures, source coding procedures which do not have to wait for such a large time but which will achieve practically the same thing in terms of efficiency, in terms of representation, efficiency of representation, right. Now certainly we will not be able to do as well as we are able to do with this ideal procedure. The number of bits that we end up using may be much more than the entropy of the source, the average number of bits that you end up using.

So the question is, what are these procedures. There are number of such procedures. These are called compact coding procedures. We will discuss one of them which is very popular, is called Huffman coding,

(Refer Slide Time 40:05)



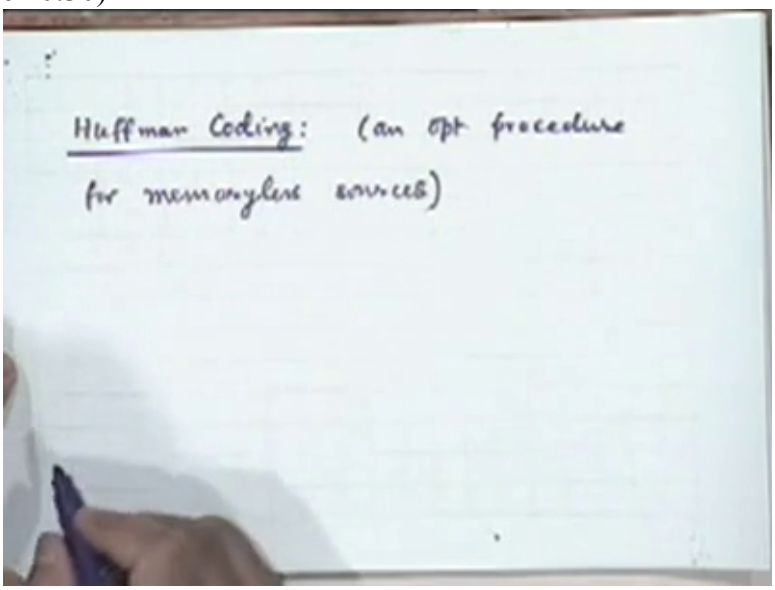
Ok. Are you familiar with Huffman coding?

(Professor – student conversation starts)

Student: Heard about it.

Professor: Ok, now you can find out what they are and in fact this can be shown to be an optimum procedure for memoryless sources,

(Refer Slide Time 40:30)

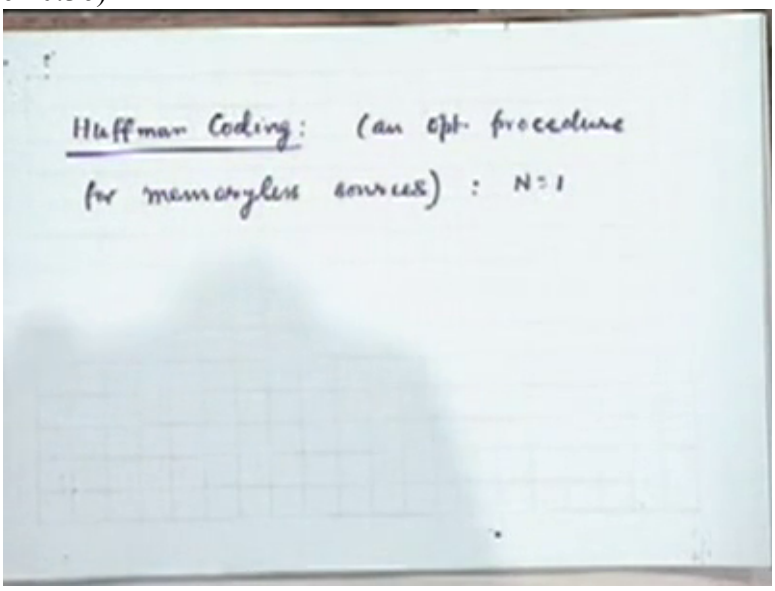


Ok.

(Professor – student conversation ends)

The basic idea is we would like to encode not with N equal to infinity but with N equal to 1

(Refer Slide Time 40:36)



that is as each symbol is emitted, we are directly encoding it rather than waiting for N symbols to be emitted and then encoding it afterwards, right?

And what does encoding mean? Encoding here means assigning to each symbol a specific digital representation, a binary representation.

(Professor – student conversation starts)

Student: Probability is a priori...

(Refer Slide Time 40:59)

Student: Sir otherwise how...

Professor: Yes, yes we will have to know that. We will have to know the probabilities. Even in that model we had to know the probabilities, right? Because unless we know the source, unless we know the statistical probabilities of source, we cannot possibly...

Student: For large number of N symbols, we can find the probability. In that case we can find out probability,

(Refer Slide Time 41:17)

we do not need to...

Professor: No, but that procedure is not about finding the probabilities. That procedure is, given the probabilities are unequal, how to reduce..., right?

Student: There was confusion like, he was saying in...

Student: In this case we can find out probabilities. If we connect N number of symbols...

Professor: Probability you will have to find. You can study that, after all you are going to look at your source very seriously if you want to use these optimal procedures. You would have offline, independently studied your source very thoroughly and have a model for that source, right? Once you have a model for that source, the question is how do I use that model efficiently. That is the limited question we are looking at.

We are not looking at the first question of how to model that source.

Student: We have modeled that source already.

Professor: Assume that you have modeled the source and we have this model available to us. Any other question?

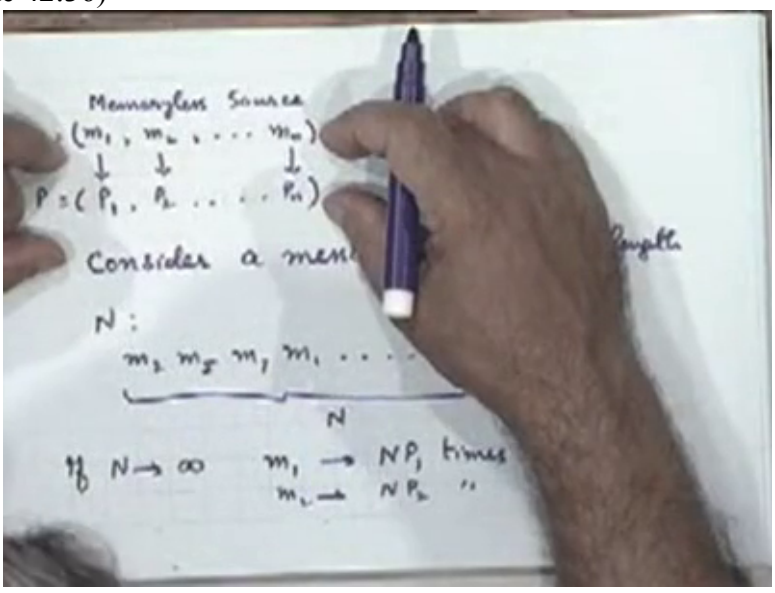
Student: Sir, when the model is available, why do we have to wait for N, means that thing...

Professor: We have to wait for N because of the fact we wanted to convert a non equi-probable message source to an equi-probable message source so that we could get efficient coding, Oh, oh I think I have probably made a mess of this lecture, then. I do not know.

(Professor – student conversation ends)

You see. I do not know of any coding procedure

(Refer Slide Time 42:36)



based on, so far when I was discussing this, I had not told you any coding procedure by which one could actually hope to achieve a symbol representation for each of these symbols which requires on an average, a length equal to H m, right? So I said I will tell you one procedure by which I could do so. How? By converting this non equi-probable message source to an equi-probable message source in which I will consider a message sequence of length N, Ok for which I already know that this can be done.

(Professor – student conversation starts)

Student: 0:43:14.5

Professor: I repeated that many times. This is the point I wanted to emphasis at that time.

Student: That was just to calculate the...

Professor: That was to demonstrate that it is possible to

Student: Convert into equi-probable

Professor: To achieve this value of representation, that is average number of symbols, average binary digits per symbol equal to entropy. To get to that result I went through that argument.

Student: Sir could you repeat what you did you said?

(Refer Slide Time 43:43)

Professor: The argument was that I want to represent this non equi-probable source by an equi-probable source such that all the probabilities are same, right, for which the result is known to me and I wanted to use that known result to evolve a coding procedure of this source, Ok. That is the objective of what we did earlier. I hope it is clear now. Ok

(Professor – student conversation ends)

Alright, it does not matter. And now I am discussing another procedure which does not require you to wait up to infinity, fine, which requires you to, which can do the coding for you as each symbol is being emitted. But we cannot hope to get the same kind of performance as that procedure gives us. Performance in terms of representation vis-a-vis entropy, right, average length being equal to entropy. We will expect the average length here to be

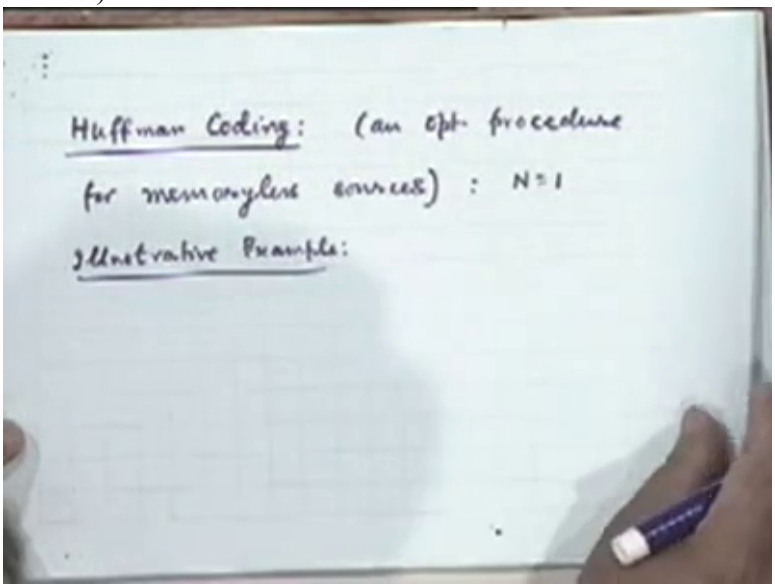
(Professor – student conversation starts)

Student: More than

Professor: More than the entropy, right. Let us see how much more it is with such coding procedures, Ok.

And I will illustrate this procedure as well as how good it is by taking a specific example, Ok. I think it is best to illustrate this coding procedure by an example. So I will take an illustrative example and that example is like this. I think I will start

(Refer Slide Time 45:09)

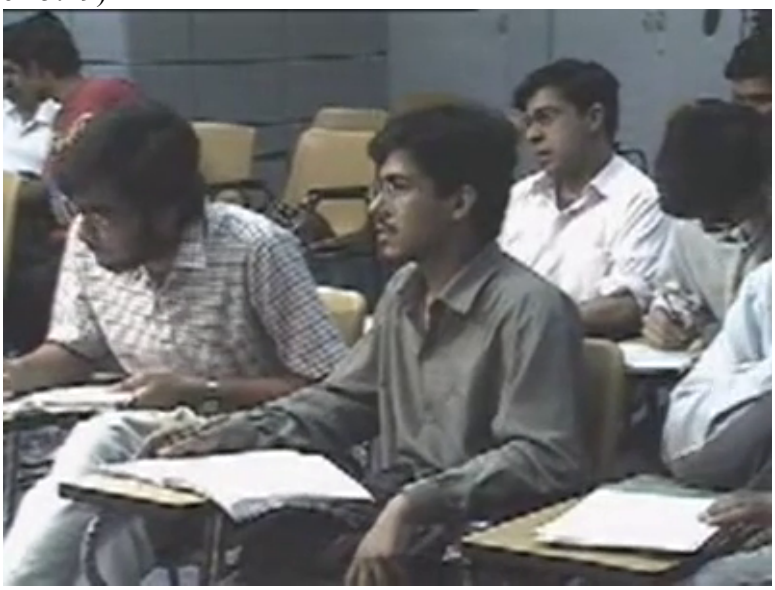


with an independent sheet.

Student: But Sir, this is the optimum 0:45:13.5 we saw already then this will...

Professor: For equi-probable sources, there is hardly any problem. Because

(Refer Slide Time 45:19)

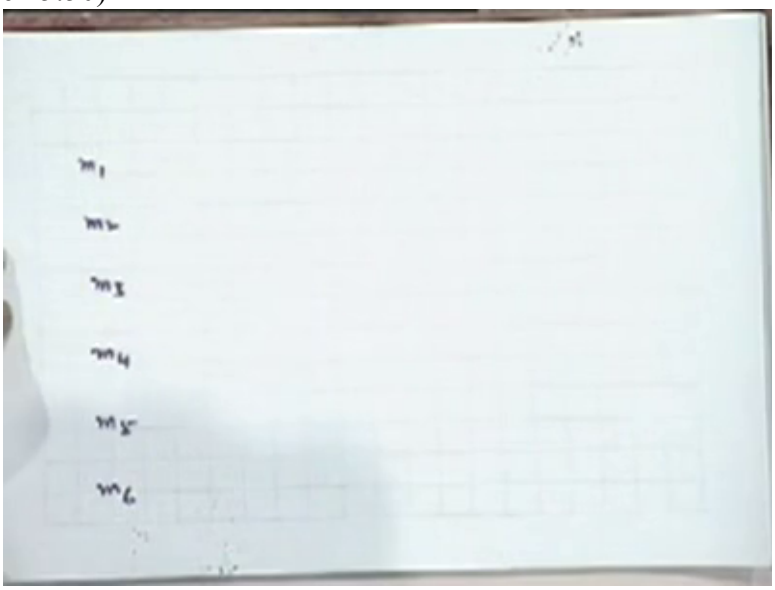


any mapping is going to come, Ok. For equi-probable sources we do not have to worry anything, we do not have to do anything special, Ok.

(Professor – student conversation ends)

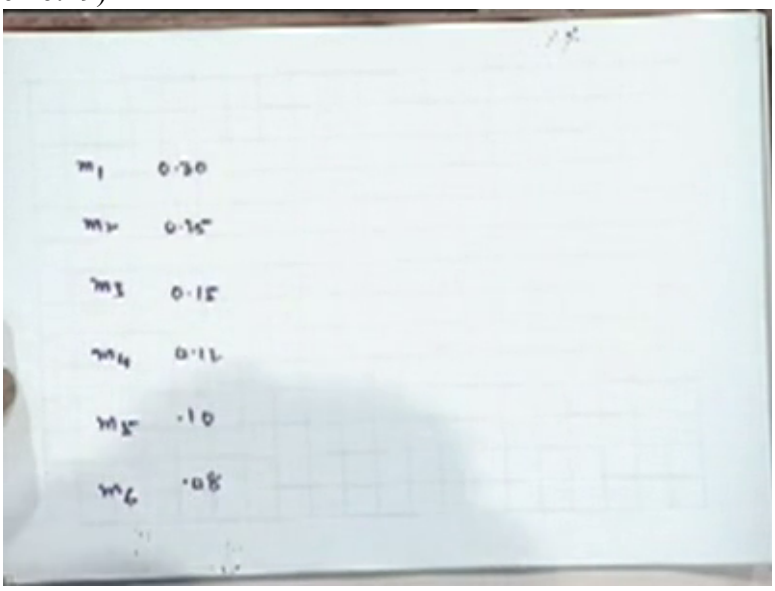
Let us say I have a source. Just I am taking this as an arbitrary example, in which the source has 6 possible symbols, dictionary of 6 symbols,

(Refer Slide Time 45:50)



Ok. And let us say these symbols are associated with corresponding probabilities which I am going to list, I am already listing them in the decreasing order of their probabilities. Let us say this has the probability of point 3, this has a probability of point 2 5, point 1 5, point 1 2, point 1 0 and point 0 8,

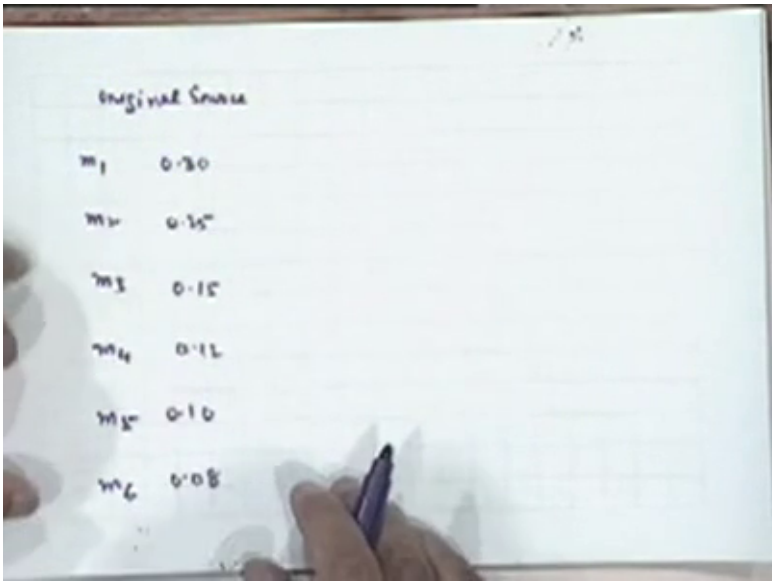
(Refer Slide Time 46:19)



alright.

I have a message source with 6 possible symbols with these corresponding probabilities. I have arranged these various symbols, message outputs in the order of decreasing probability of appearance, Ok. So this is my original source. This is the model for my source. These are the messages

(Refer Slide Time 46:49)



which it emits, which it can emit in any unit interval with these probabilities, right? Now what I do is the coding procedure proceeds like this.

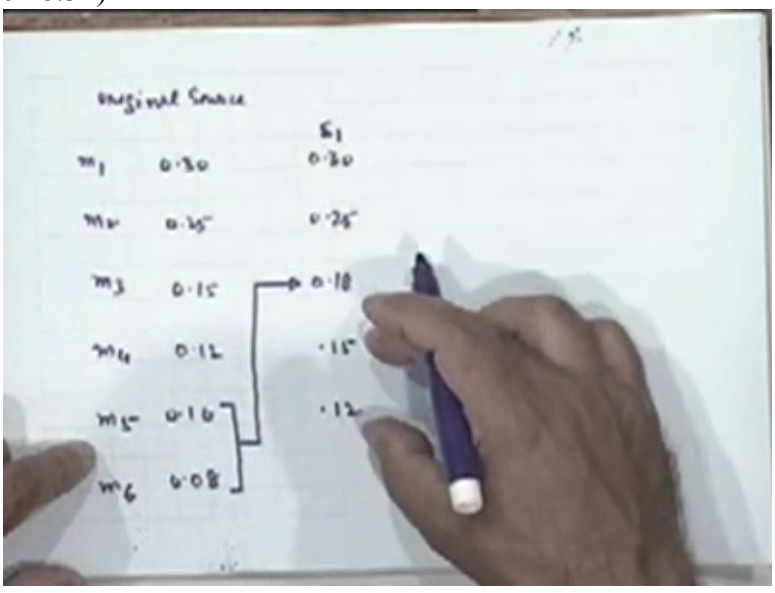
I construct a new equivalent source S 1 with only 5 messages by doing the following, by clubbing two of the lower, lowest most messages into a single message, single equivalent message. So I am constructing a new message source in which the dictionary is comprised of these 4 and the fifth one is the union of these two messages that is either m 5 or m 6. I am counting that as a single message, right?

Alright and then I look at this 5 message source and again put the probabilities in the decreasing order. This new message, since they are independently emitted, what is the probability of this new message value?

(Professor – student conversation starts)
Student: point 1 8
Professor: Point 1 8 because it is the union of these two events and each of them is independent, not really independent I should use mutually exclusive.
Student: Mutually exclusive
Professor: Right.
(Professor – student conversation ends)

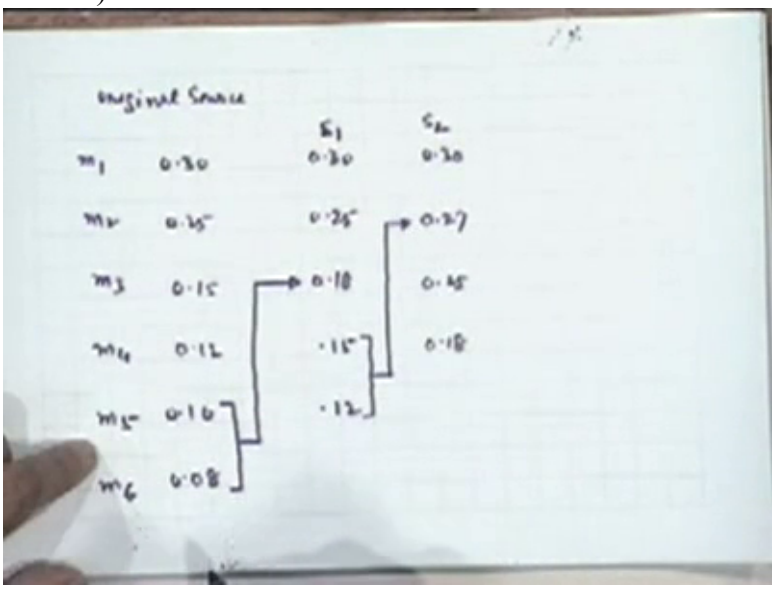
So this will be point 3 0. Then point 2 5 and here we will write point 1 8, point 1 5 and point 1 2. So these two together have been clubbed into a single message and that message has appeared here in this. And I repeat this procedure.

(Refer Slide Time 48:34)



Again I take the lowest two symbols, club them into a single symbol and construct a new message source which will become point 2 7, so it will come here, point 2 5, point 1 8, is that right? Fine?

(Refer Slide Time 49:04)



And I keep on doing this.

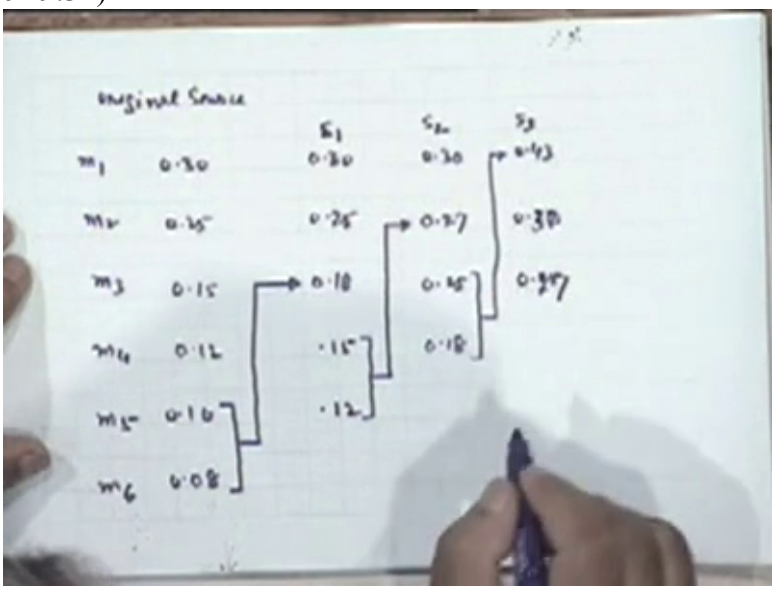
Now I will get point 4 3 which will be the largest, then point 2 7 and then point 2 5, correct?

(Professor – student conversation starts)

Student: Point 3, point 2 7.

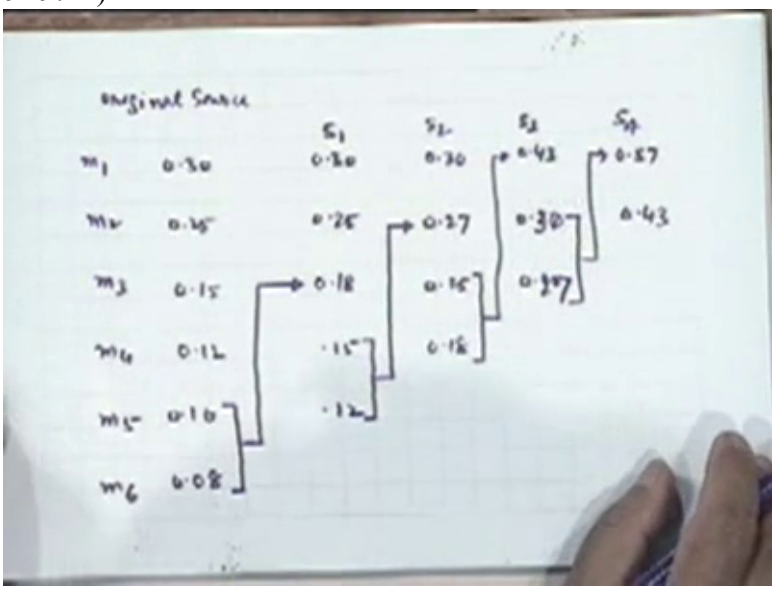
Professor: Oh, point 3, point 2 7,

(Refer Slide Time 49:31)



fine and this where I

(Refer Slide Time 49:44)



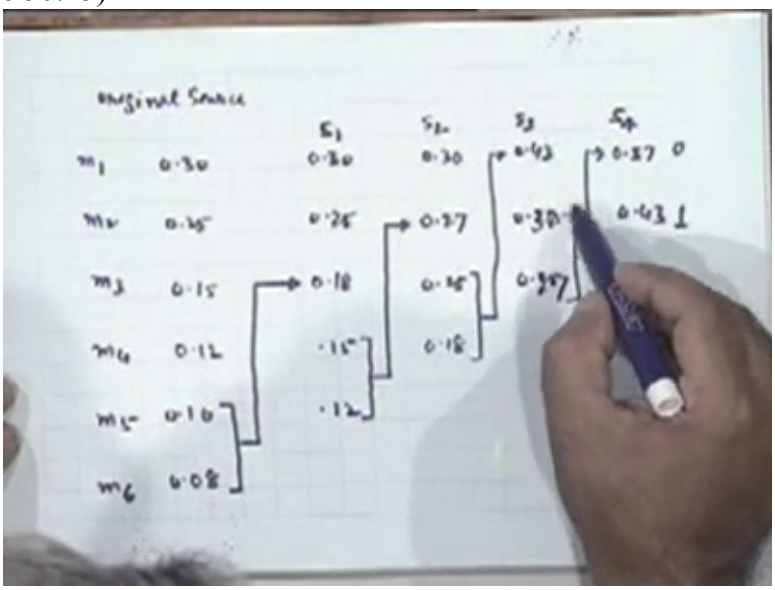
stop. I stop when I am left with only 2 possible emissions, right? What do you think I have hoped, what did I hope to achieve by doing this?

(Professor – student conversation ends)

I, basically the objective was to finally obtain a source of only 2 messages with as equal probabilities as possible, Ok. That was the basic idea, and this procedure helps me to achieve that objective, Ok. And if these are as equal as possible, it will be reasonable now to do a coding procedure as if you are doing, you are dealing with equi-probable sources, right? I assign arbitrarily a code zero to this, and a code 1 to this, or other way round, it does not really matter, right?

And now I proceed backwards. I trace the path backwards. I have to now distinguish between the two

(Refer Slide Time 50:45)



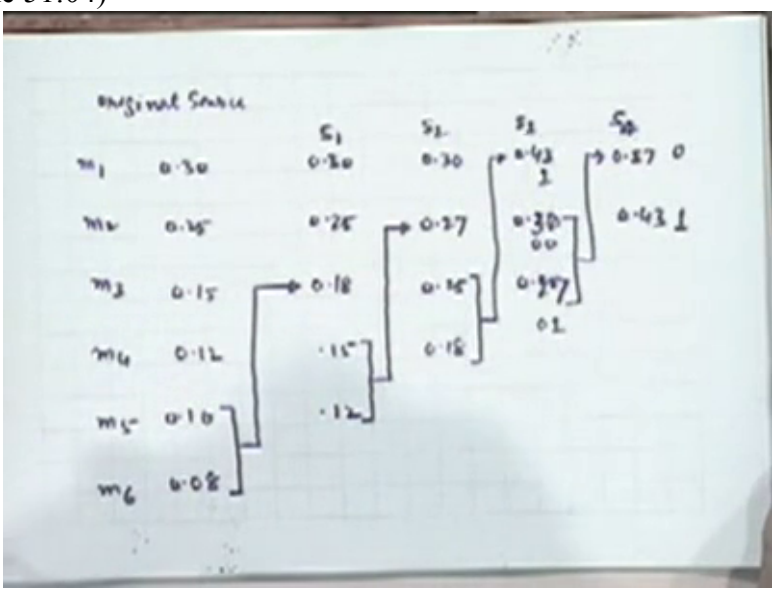
messages that comprise this message. This message is actually comprised of the union of these 2 messages which eventually I have to distinguish between, right? So I represent this with

(Professor – student conversation starts)

Student: Zero

Professor: A zero zero and this with a zero 1, and this of course remains 1.

(Refer Slide Time 51:04)

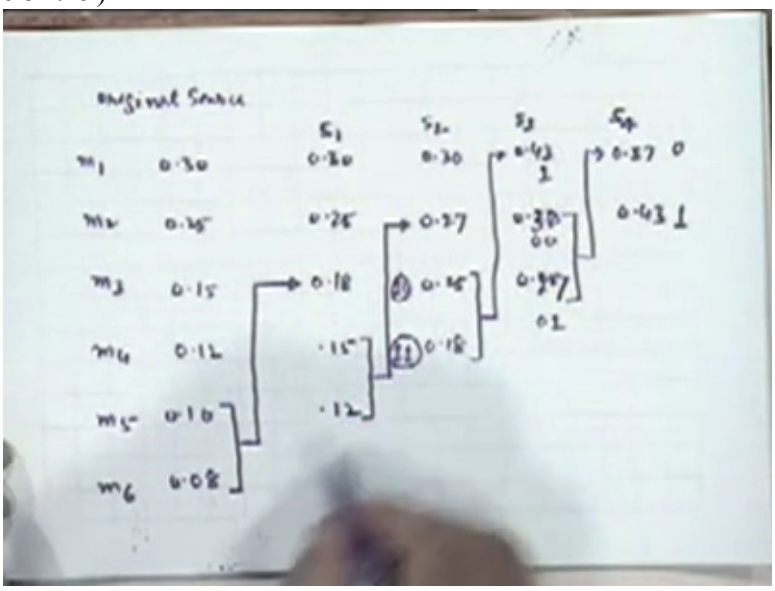


Because I do not have to do any distinction here.

(Professor – student conversation ends)

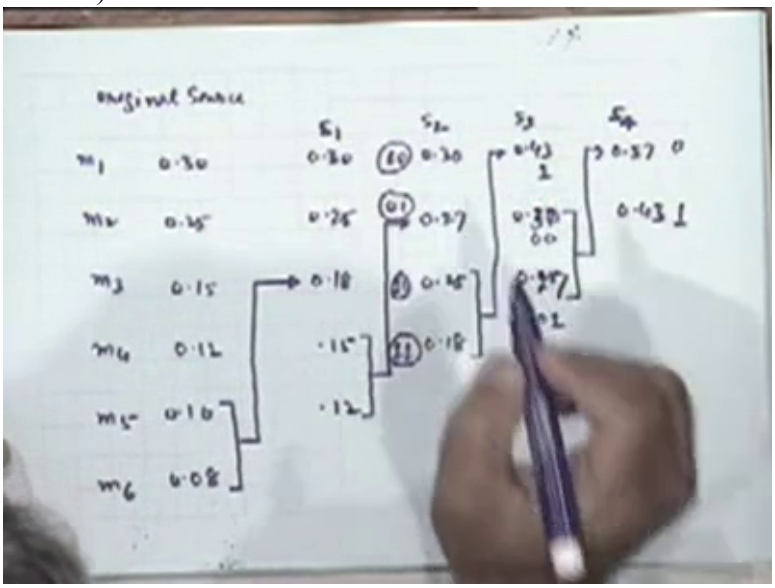
But when I go one step backwards, I have to now distinguish between these 2 symbols. So I write here a 1 zero, and a 1 1. And this becomes

(Refer Slide Time 51:19)



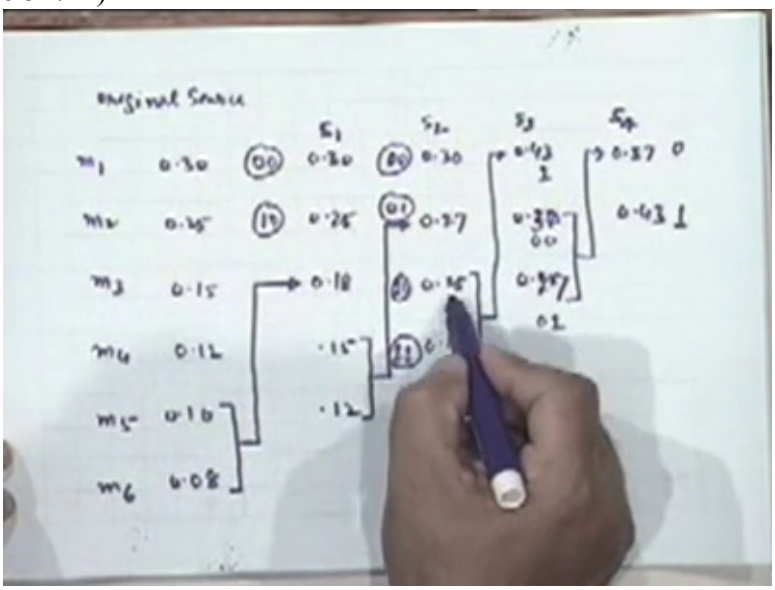
zero zero and this is zero 1,

(Refer Slide Time 51:26)



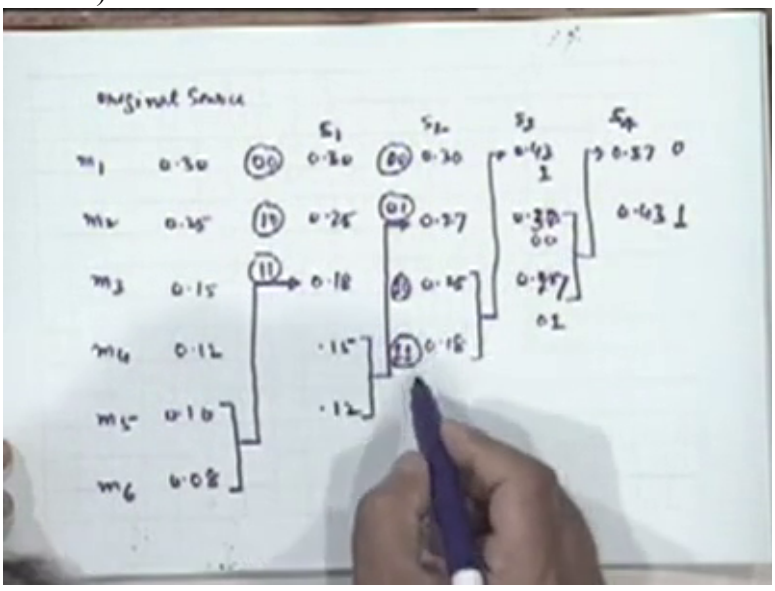
Ok. Keep on doing that. This will be zero zero, this will be 1 zero right, point 2 5,

(Refer Slide Time 51:41)



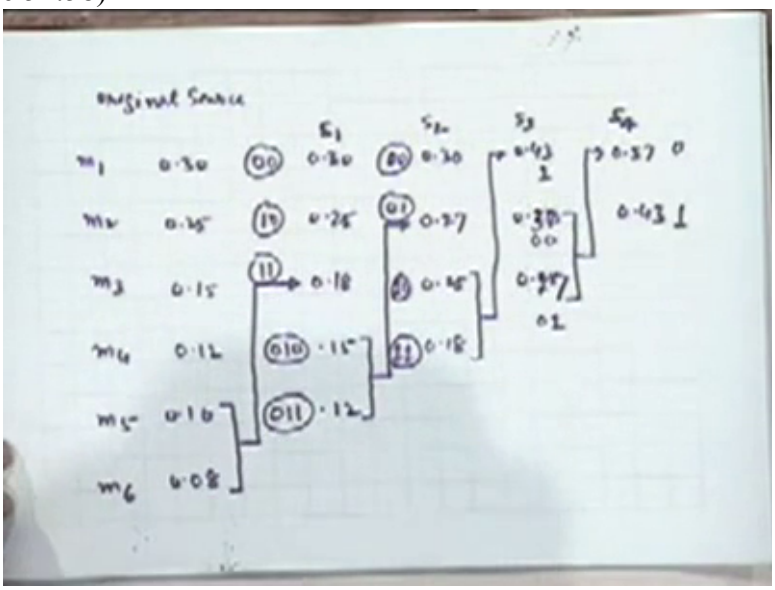
point 1 8 is 1 1,

(Refer Slide Time 51:46)



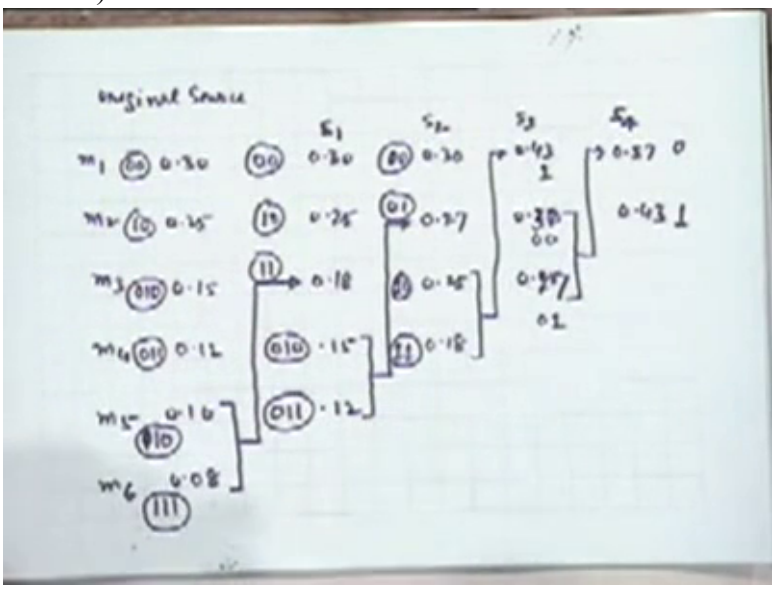
and this has got to be again distinguished between these 2 symbols, so I write here zero 1 zero and zero 1 1,

(Refer Slide Time 51:58)



Ok. Fine, go to the last step. You get zero zero here, 1 zero here, this was zero 1 zero, this was zero 1 1, and this will become, yes 1 1 zero and 1 1 1.

(Refer Slide Time 52:33)



And this is your final mapping by Huffman coding.

m 1 will be represented by a sequence, a binary representation of zero zero, m 2 with 1 zero, m 3 with zero 1 zero, m 4 with zero 1 1 and so on, right?

(Professor – student conversation starts)

Student: Sir this is the problem I was saying, in a signal,

(Refer Slide Time 52:55)

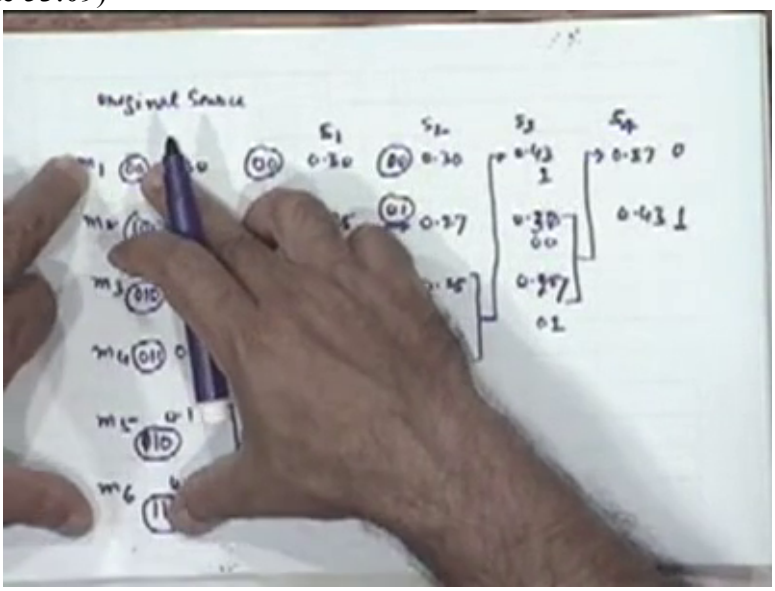
let us say, signal of N symbols. How will we decide that it is in fact...?

Student: All three bits

Student: Encoded...

Professor: No, there we are not. We are not; we are not doing encoding on a single symbol basis.

(Refer Slide Time 53:09)



We are encoding on N symbol basis.

(Professor – student conversation ends)

We have to distinguish between one sequence of N symbols and another sequence of N symbols, and other sequences of N symbols, so each unique sequence of N symbols will have

a unique representation, binary representation. That is all. So I will know the specific sequence that will be represented by it.

(Professor – student conversation starts)

Student: How do we know that zero 1 zero, is a zero followed by 1 zero...

Professor: Ok, that is a separate question but has this question been answered? Tarun?

Student: Yes

Professor: Have you understood what I am saying, trying to say? In that procedure, I am mapping this whole sequence by a corresponding binary sequence and I will also have to decode also in the same way. I have to wait for the whole sequence to be received and then decode the whole sequence. It cannot be decoded on a symbol-by-symbol basis. One has to wait for the infinite sequence to be encoded, this infinite sequence to be transmitted, received and then you have to decode that as a whole sequence together, not on a symbol-by-symbol basis.

But in this case, I can do on a symbol-by-symbol basis. I can look at, you know, 2 bits and say that this is m 1 or say this is m 5.

Student: Sir that is what my question is. How do you decide whether to look at 0:54:20.7?

Professor: Oh, was that your question?

Student: Yes Sir

Professor: I thought you were referring to the previous procedure.

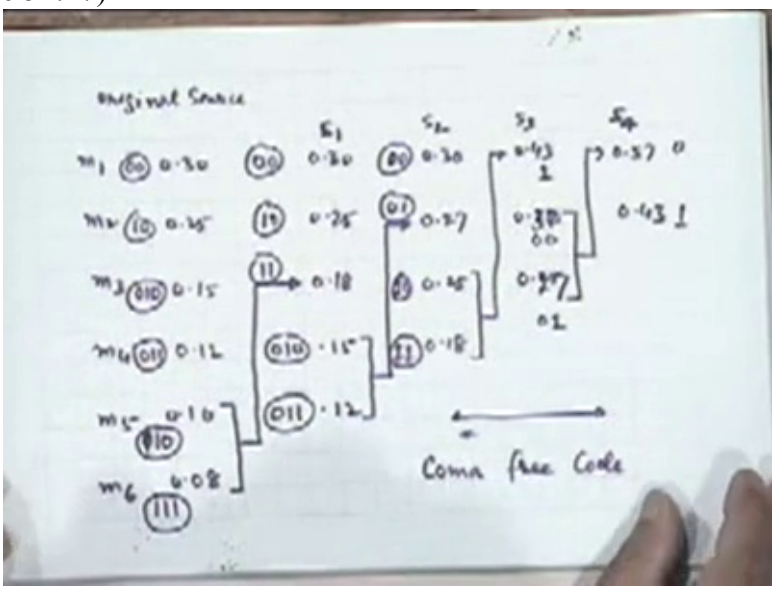
Student: Here like when we talk...

Professor: Ok. Then yours and Varun's question was the same, Ok.

Student: I am saying that if it is zero 1 zero, how do we know that it is zero...

Professor: Ok, fine, fine. That question has a different answer. The answer to that question is it can be shown that that the Huffman coding procedure is, is what is called a comma free code. Right

(Refer Slide Time 54:47)

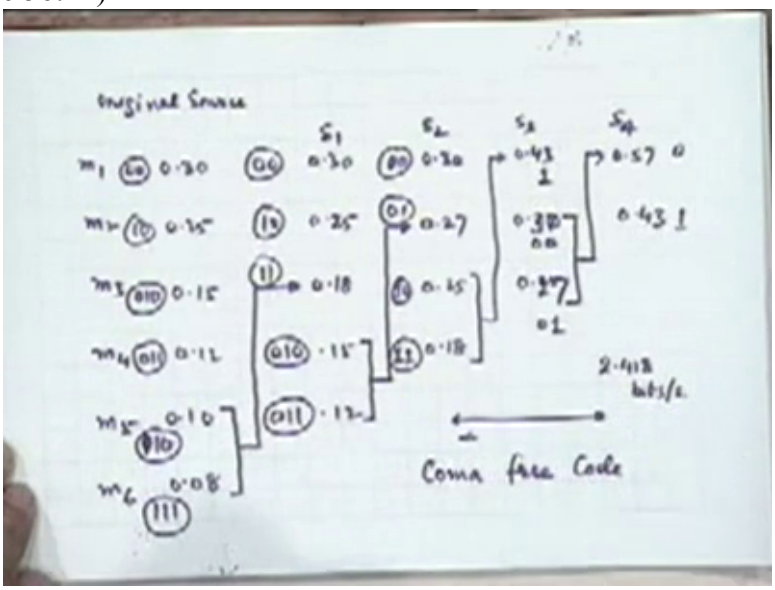


that is one does not require a marker between code symbols. See to do that, one way would be to send a marker in between that, right? So a specific symbol which says that Ok, one codeword is over and next codeword is starting. But that will be inefficient because you are again using a marker space. It can be shown that given a particular received sequence, if there is no error, Huffman code can be uniquely decoded, Ok.

(Professor – student conversation ends)

That is, there is no other way of allocating sequence of bits to the original message sequence other than a unique way, fine. That is, Ok, I think the best thing is to construct a simple example and see that you cannot indeed cause any other allocation except the right allocation, right? I would like you to do that as an exercise. It is a very simple exercise. In any case our time is up now. But before I finish, let me make a simple comment on the efficiency of this code, how good this coding procedure is. For this specific situation, suppose I was to compare it with entropy, entropy you can calculate easily, given the probabilities? If you do that, I will just give you the value. You can verify it. It is 2 point 4 1 8 or something, bits per symbol

(Refer Slide Time 56:14)



(Professor – student conversation starts)

Student: This is the entropy?

Professor: This is the entropy corresponding to these probabilities. You can also compute the average length. How?

Student: 0:56:22.3

Professor: This will be 2 into point 3, 2 into, plus 2 into point 2 5 plus 3 into point 1 5 plus 3 into point 1 2 plus 3 into point 1 plus 3 into point 0 8. This is the average length for the final code that we have achieved. And that turns out to be 2 point 4 5, right.

Student: Greater than...

Professor: Obviously it has to be greater.

Student: But it is very close

Professor: But it is very close, right. In fact one can define what is called as code efficiency as H m by average length. That is very, very good, that is nearly point 9 7 6 or something. I will start from this point next time, right? I hoped to do this plus much more but thanks to our confusion in between we could not do that.

(Professor – student conversation ends)