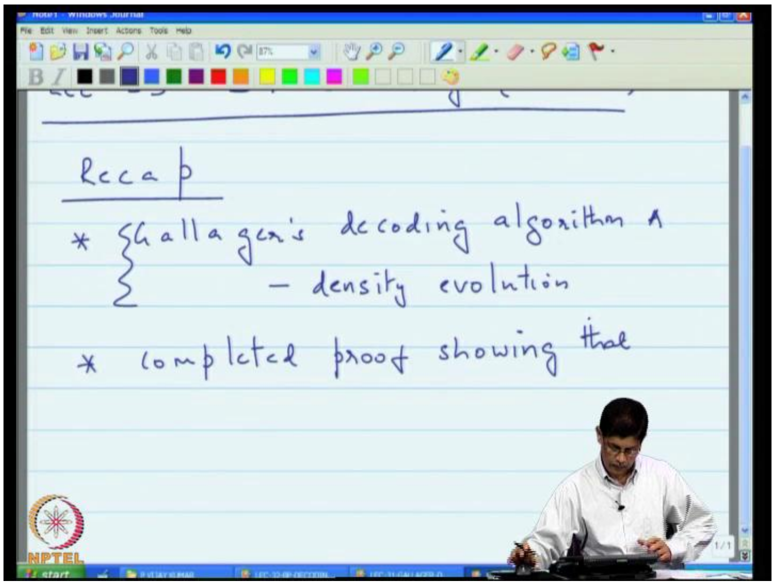
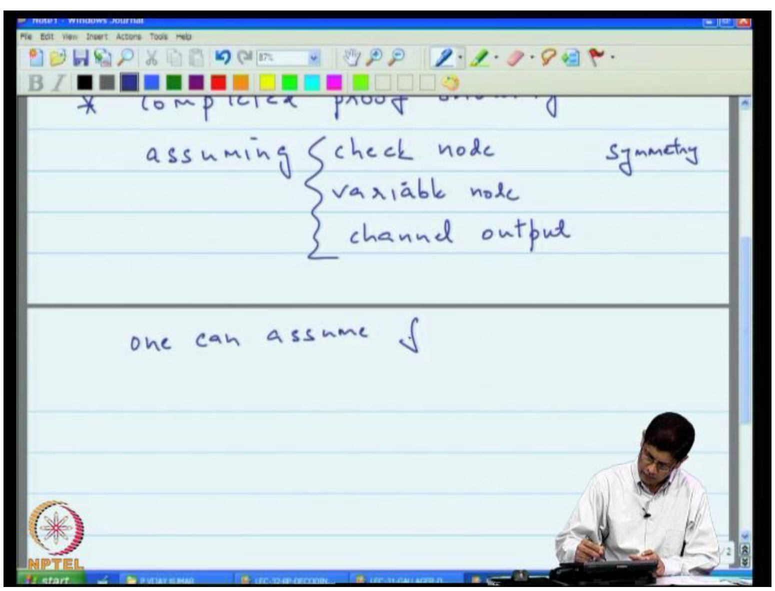
**Error Correcting Codes**
**Dr. P Vijay Kumar**
**Electrical Communication Engineering**
**Indian Institute of Science, Bangalore**

**Lecture No # 33**
**BP Decoding (Continued)**

(Refer Slide Time: 00:23)

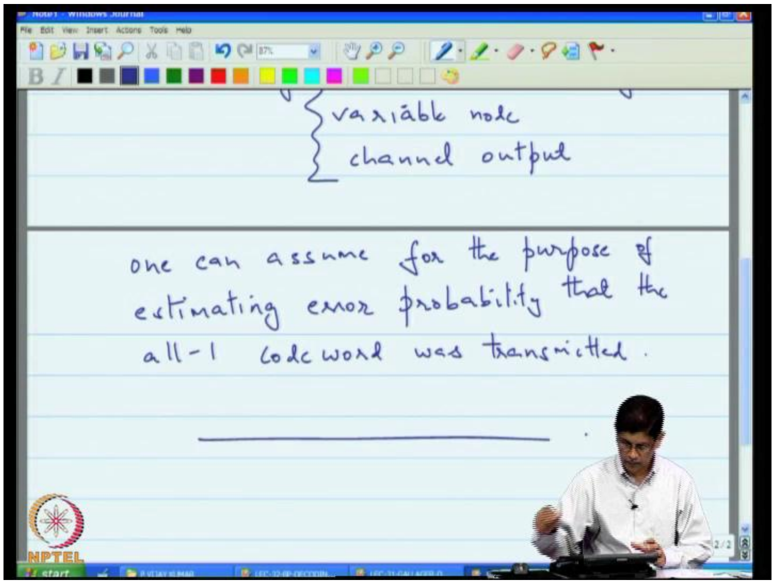



(Refer Slide Time: 01:36)



Welcome back, we are on to lecture thirty-three now. And let me begin with a quick recap; last time we looked at Gallager's decoding algorithm A, where we looked at density evolution, and then actually proved the statement that I had made earlier completed proof showing that, showing that assuming check node, variable node and channel output symmetry.

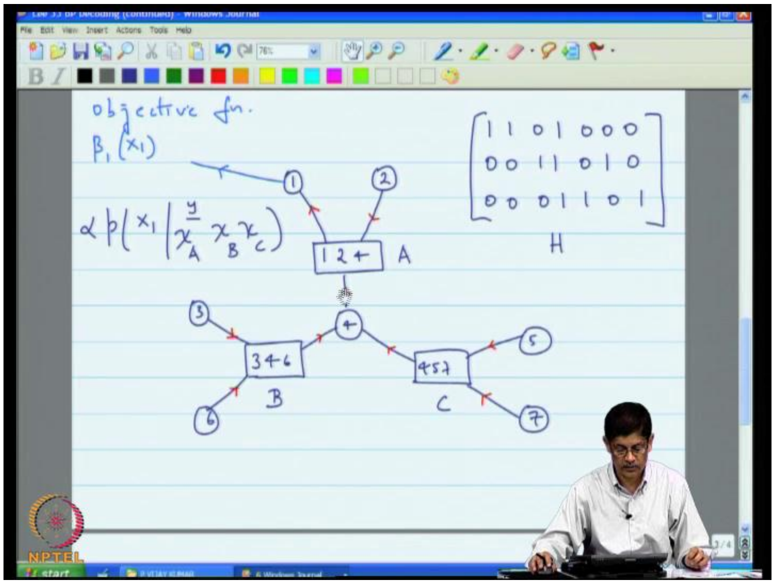
(Refer Slide Time: 02:14)




One can assume for the purposes, for the purpose of estimating error probability, that the all-one code word was transmitted. And today, what I would have actually like to do, so the title for our lecture today is, more or less, the same as the last time. We did not quite get the belief propagation decoding, BP here stand for belief propagation, so we did not really get the belief propagation decoding of LDPC codes. So, what I have done is, I have carried over the title. So, let me just make sure, that I give this lecture a title .

So, what do you mean by belief propagation decoding and I am going to actually… So, I want to tell you first of all, that belief propagation decoding is no different than the method of decoding that was employed, employed on the junction tree. So, I would like to bring that back and then I will continue our discussion. So, let us see if I can pull that up from one of our previous, so this will actually suffice although I may have another lecture, which describes the…

(Refer Slide Time: 05:24)



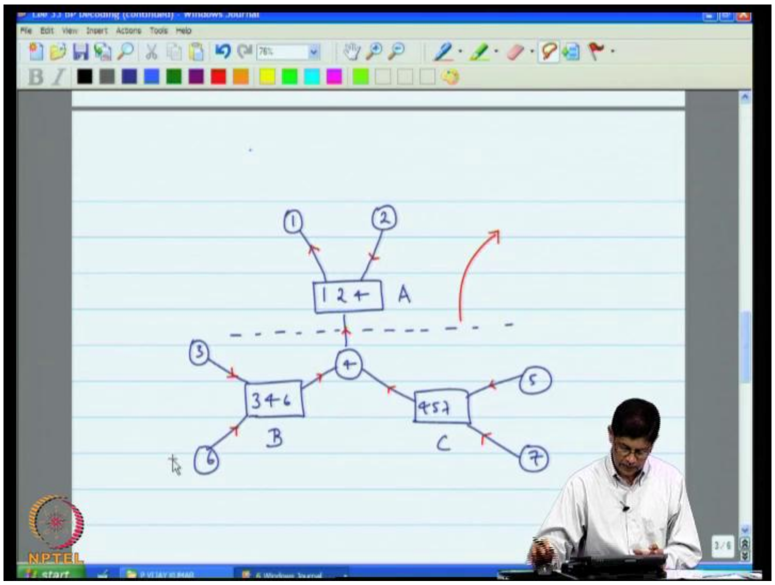

So, let me just pick this particular page, put it into our current lecture. So, let us assume, so this is our 7 4 2 code block code, that we actually looked at early, which had associated to it a certain parity check matrix. And let us assume that we are back in which we posed the decoding problem, the maximum likelihood code symbol, decoding problem as an MPF problem and then proceeded to actually solve it using the GDL.

Now, we know that in the GDL you first organize a local domain into a junction tree and then, after you have the junction tree, then your next step is to actually pass messages on junction tree according to a certain schedule. Let us assume, that for the time being we are interested in computing. Let us say we are interested in the single vertex problem of the GDL, in which case, we would orient all the edges towards that particular vertex. And at the end, after we passed on messages and also, there is local at every node and eventually, we would be interested in computing the objective function, the objective function, which in generic notation is beta 1 of x 1. But our notation is, basically, a quantity, that is proportional to p of x 1 given y, where y is entire received signal.

Now, I want to actually modify this expression slightly to reflect the fact, that we are, we are computing this probability based on the received symbols. Now, the other fact, that this graph takes for granted is that this computed assuming, that you know the received signal and also assuming, that the seven code symbols, original code symbols were really drawn from a code corresponding to a certain parity check matrix. To emphasize that let me just put the parity check matrix back here. So, this is the parity check matrix  of the code and there are three parities. The parity corresponding to the first row, the second row and the third row and when we computed this, this was, conditioned the, computed taking this into account.

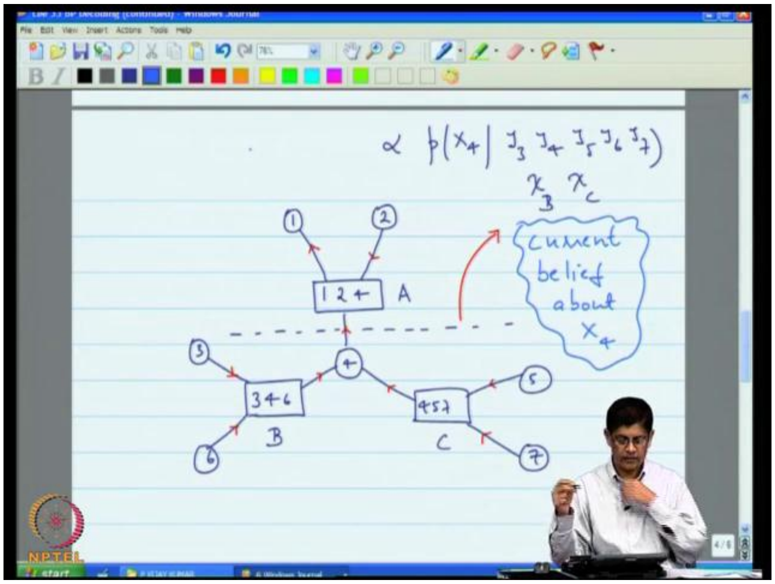
So, to to make that little bit more explicit, let me just rewrite this and saying, that this is, this is p of x 1 given the entire received symbol. But also given three parity check, which is kai a, kai b, kai c where I am going to refer to this parity check as A, this is B and this is C, and I think I have done this before. And the quantity that we have computed is not exactly equal to this, but is proportional, which for our purpose is actually good enough, is proportional to this, so this is in objective function. Now on the other hand, supposing so we know what this message is, but if I stopped you at some point here and asked you, by the way, would you be able to tell you, what is the message, that is actually being passed here that is, from variable node four to this check node.

(Refer Slide Time: 10:20)




So, let me copy this picture over, so that I will use that to make that point. So, this time, let us say, that my focus is on the message, that is going on over here, not so much on the objective function there and we already know, that we are working with this particular code. And I ask you what is the message, that is actually being passed across this barrier, across this edge and it is not hard to actually show, that the message, that is passed here across this. And since I am going to do this a couple of times, let me do the following, let me copy this page a couple of times over. So, across this, what I am actually computing? You can show in analogous what you are actually computing is p of x 4 given, given some evidence.

(Refer Slide Time: 11:20)

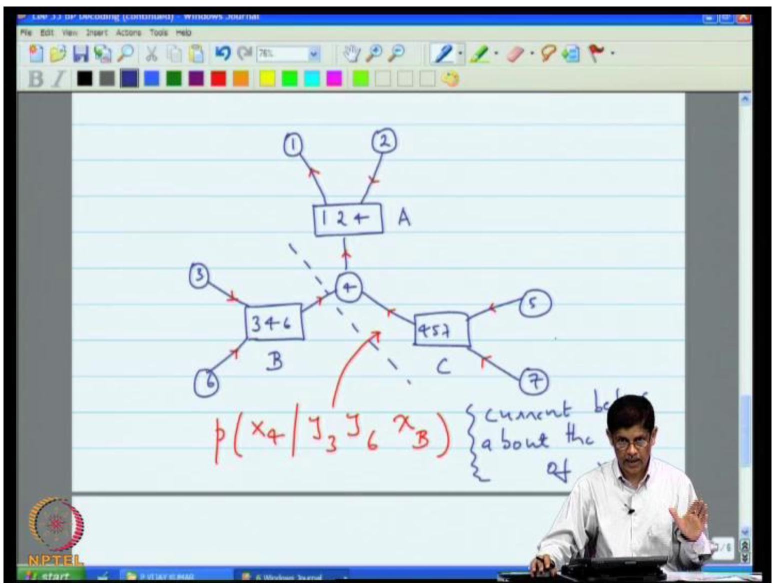



So, this node, this is passing on its belief about x 4 given all the evidence that it has received. In particular it has received information about y 3 y 5 y 6 y 7 as well as y 4, and in addition it also knows about the presence of these two parity checks. We can actually say that the message that has been passed is proportional to x 4 given, y y 3 y 4 y 5 y 6 y 7. It is also being given knowledge of presence of these two parity checks. So, given this, it is passing on. So, what you might do is and, and how do you ok, now you might ask, well how do you get to actually show that?

So, if you actually try to formulate, this is an MPF problem and you try to, just the way as we formulated the problem of code symbol, decoding of x 1 as a MPF problem. If you do that, then you will come up with the junction tree, which is precisely the proportion of this junction tree, which is below the dotted line. So, this is what you will come up with and you will you find that then you are actually computing an objective function, which is precisely this. So, that is the good way to look about, look at it.

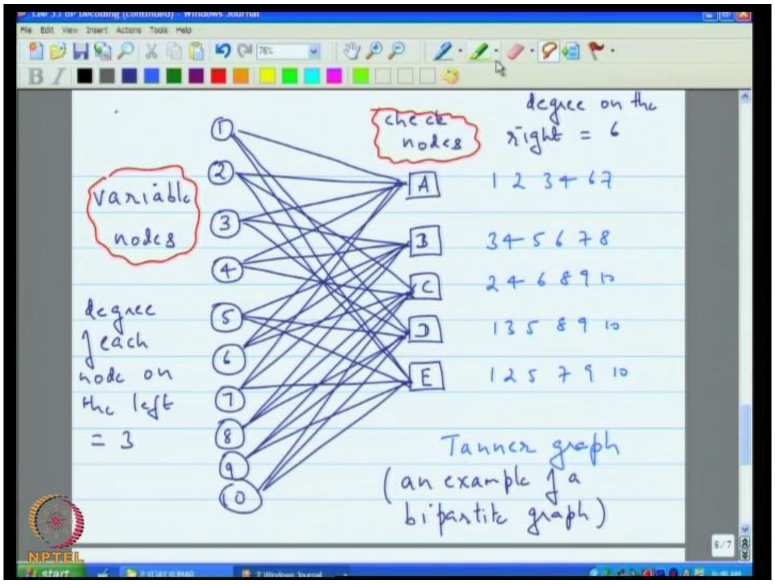
So, if you actually want to see what the message is, that is passing across any node, you can just isolate the path, that is before that from which messages are being passed. And say, look I understand what this little network here is doing because it is the junction tree set up, solve it, it is the, there is an overall junction tree, but it contains smaller junction trees each of which is doing a similar computation. So, the this junction here, basically what you are doing is you are computing the probability of x 4 given the evidence, which in this case is y 3 y 4 y 5 y 6 y 7. And in a way we have evidence, additional evidence, that the 3 4 6 symbols satisfy parity, as do the 4 5 6  symbols and you can interrupt this as a belief. This can be regarded as a, as the current belief, is the current belief about the message symbol X 4, that was being passed.

(Refer Slide Time: 13:56)



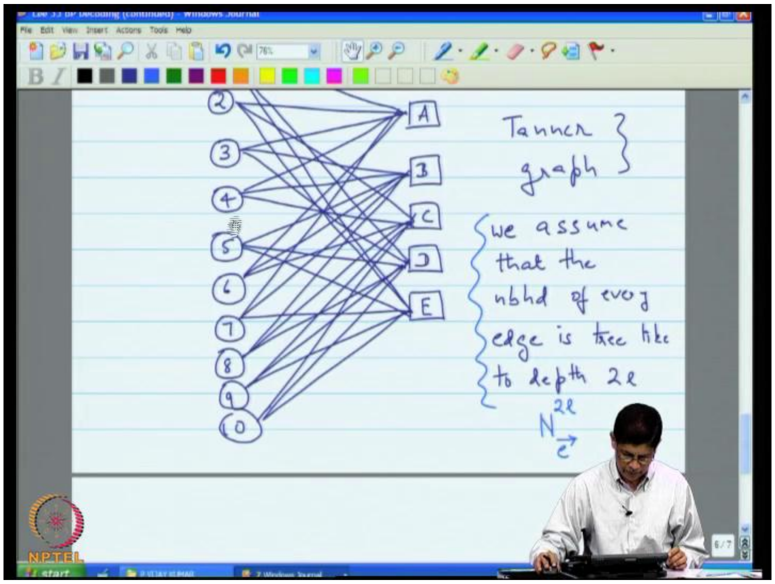

And now, what, let us look at another just to make this one more example. Supposing, I now were to stop you and ask you, well, can you tell me what message is going across, let us say this barrier over here, what is the message, that is going across from node B to node 4. And once again you can actually show that the massage, that is going across here, the message, that is going in this direction, like this is your belief about x 4, so it is p of x 4. What you are given is the evidence, that you are given is y 3, you are given y 6 as well as you are given the fact, that the 3 4 6 satisfy parity check. So, again the message that you are passing has an interpretation as your current belief. So, once again this is your current belief in the probability of x four, alright. So, now in this context we understand, that the messages that are being passed are passed across the network, have an interpretation as beliefs. So, that is clear. Now, we want to actually go on, move on to talking about LDPC codes. So, let me, just to give you a sense of the context, let me pull up our one of our early lecture on LDPC codes.

(Refer Slide Time: 16:44)




So now, supposing we are now working in the context of LDPC codes and I want to talk about the message that are being passed and interpret them as beliefs. So, the question is, can I really extrapolate from what I saw on the junction tree to this more general setting and I want to make that connection. So, if you are this is just to give you the setting.

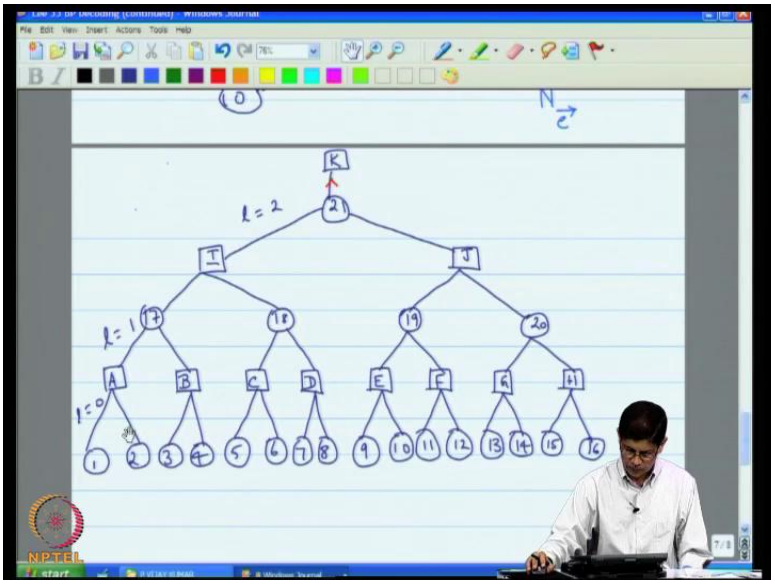
(Refer Slide Time: 17:28)




So, now, we are let us say, working with LDPC codes, and this is the Tanner graph of an LDPC code. So now, what I would like to do is to say, supposing actually unravel the Tanner graph, that is, that I am passing messages back and forth from left to right and right to left in this Tanner graph. And what I want to do is I want to actually view it, unravel this and get the better understanding of this. For this I am going to assume, that now we went through some definitions about neighborhoods, etcetera.

So, I am going to assume, that the neighborhood of every edges to death 12 is tree like. So, let us put that assumption down and we are going to say, that this is true to depth 2l. So, if, if, if it is, so their notation for this was N sub E for the l and let us make that directive and then we would have a 2l. So, assume that this is tree like.

(Refer Slide Time: 20:15)



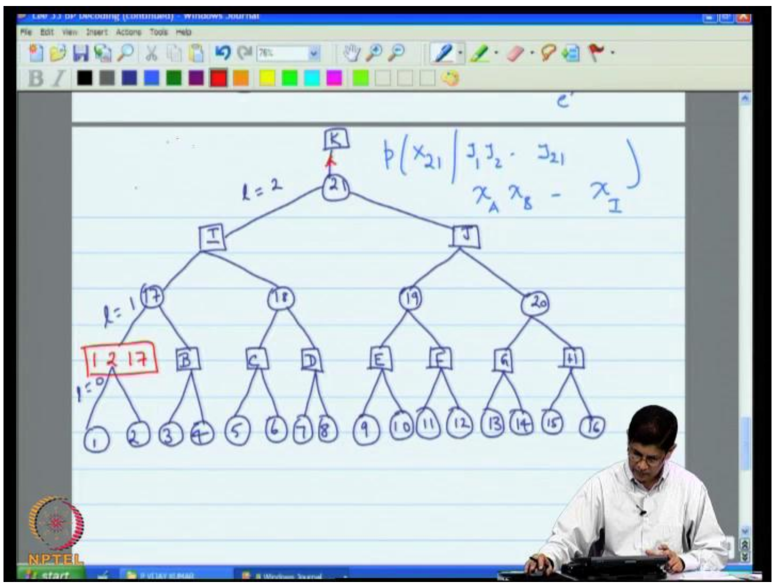

Now, I am going to, this edge could either be going from variable node to check node or check node to variable node. In this particular instance, let me consider unraveling it from a particular variable node and if you draw this out. So, let me see if we have such a figure available previously; here is the figure I was looking for. Alright, here I go, let me remove; delete some of these comments, that we would not need here today, alright. Now, I am thinking of this figure here, that I think we have done before. I am thinking of this graph here as unraveling of this, that is, I am looking at a particular iteration.

Let us say, I am looking at, so here what we have is, in this figure you have a 0th iteration where you are, actually, passing message from the variable node to check node. Thereafter, you start the first iteration and this is the beginning of the second iteration, and in the second iteration at the second face of this each iteration of the face. The first phase being when you pass message from a channel node, from the check node to variable node, and in the second phase, where you actually go backwards from the variable node to check node.

So, here now I am interested in particular in message that is going across, let us say, over here. So, this is from, this is from variable node to check node and let us pretend, that we are looking at the second phase of the second iteration, right. So, the first phase, actually, started here, the second phase is that and we will assume, that the neighborhood is tree like, so that all these of this particular edge is tree like. And I guess to depth four, which means, that all the nodes are distinct, as show here.

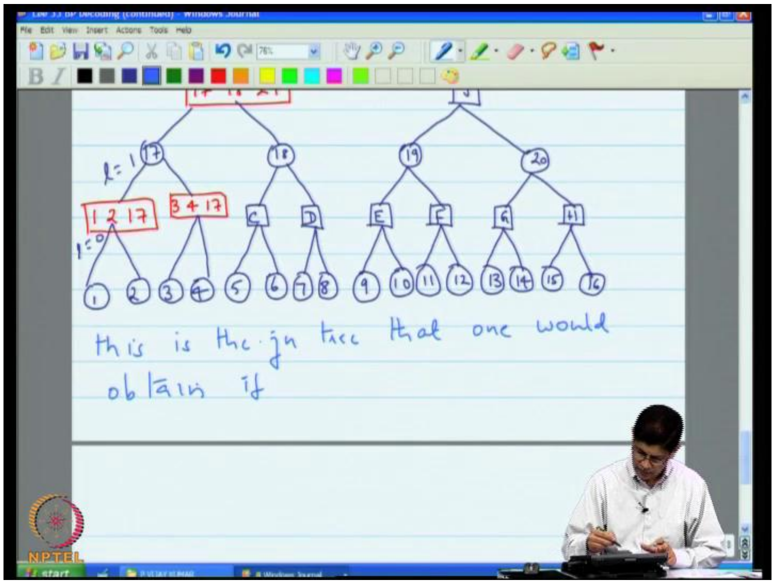
Now, what is the relation, from this, between this? What we have been discussing, what I want to actually say is, that under the assumption of the tree like neighborhood, you can actually view this as a junction tree. In fact, it is a junction tree, it, it is the junction tree. In fact, not only it is a junction tree, it is the junction tree, that you will, you end up if you are trying to, actually, set up the problem of code symbol decoding of this particular symbol here.

(Refer Slide Time: 23:21)



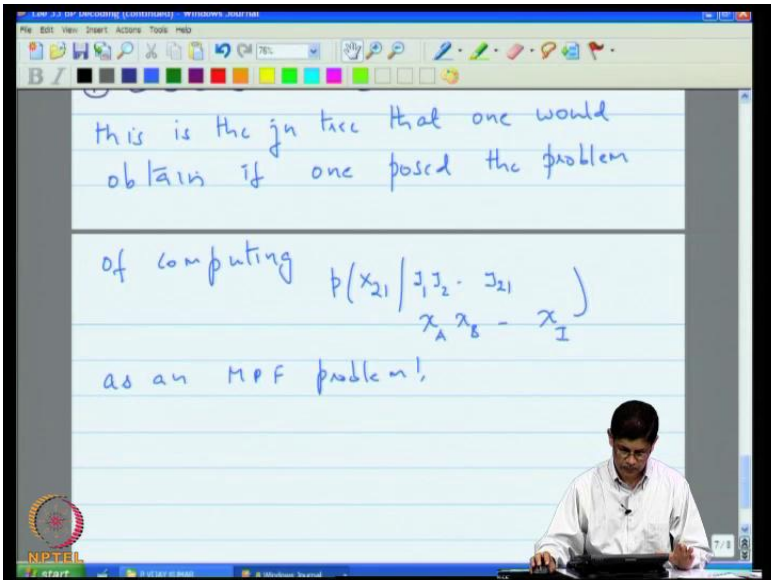
So, I said this is the junction tree. What is that junction tree computing? Let me just redraw that little bit better first. The function, that we are attempting to compute here is, is the belief p of x 21 given, given what you are, given all the received symbols: y 1, y 2, all the way up to y 21. We are also given all the check symbol ranging from A, B, all the way up to kai sub I. So, this is the belief, that you will actually compute and in fact, I do not have to prove anything to you because it is already clear from the manner in which we actually discussed the junction tree of the block code. The only thing, that one might have to point out is, that this is the junction tree.

(Refer Slide Time: 24:27)



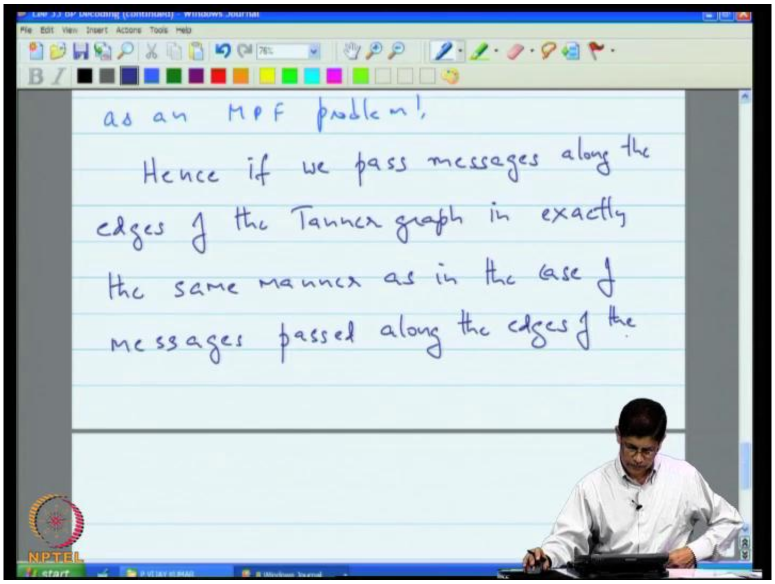

Why is this junction tree? Why this is a junction tree? Because what I could do is, that I can actually rewrite the check node slightly differently. For example, this check node over here, I can write it as check node and I will write it in red so that stands out, 1, 2, 17 and this check node here is 3 4 17. So, now, if you actually replaced all the check node where I have actually introduced letters by this symbol, then you actually see, that this is a junction tree because now, supposing you project on to the code symbol on to the variable 2, which transfer x 2, then you can see, that the projection is connected. Similarly, if you project on to 17 you will notice, that actually, this one here with be replaced by so let us replace this as well, this would then be replaced by 17 18 21 and you will have this. And now, if you actually project this entire graph on to 17, then you will see that 17 will show up here, here, here, and the projection is actually connected.

(Refer Slide Time: 26:13)

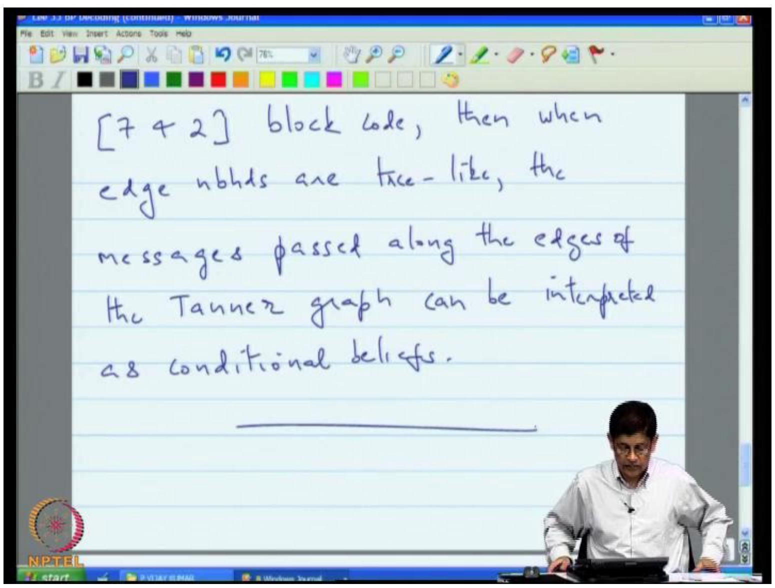



The conclusion is, that this is a junction tree, so this is the junction tree that one would obtain if, if one posed the problem of computing, of computing this quantity here, of computing this quantity as an MPF problem. So, what that means is, that if in the Tanner graph, if in the Tanner graph we were to pass messages, this is important, perhaps I should have said earlier, if in the Tanner graph we were to pass messages in just the same way as we passed messages on the junction tree of our earlier example block code, then assuming, that neighborhoods are tree like, then you would have the same situation as in the block code and the messages, that you passed could be regarded as beliefs.

(Refer Slide Time: 28:17)

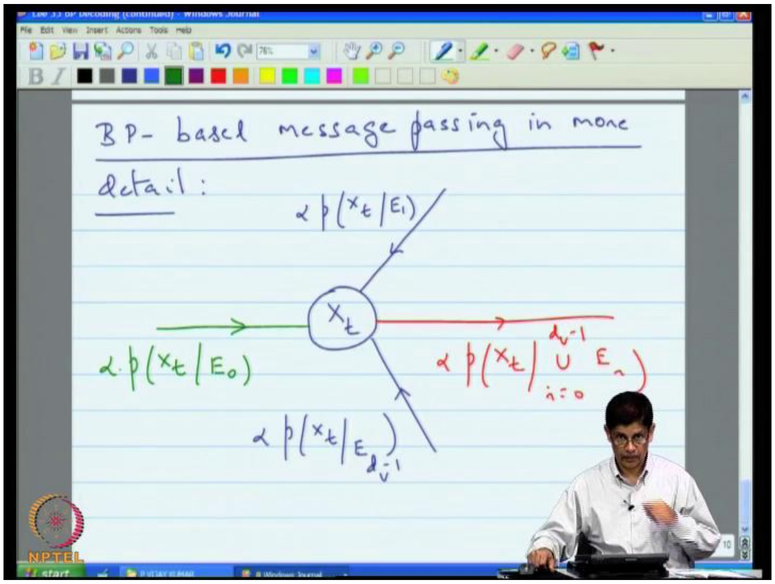



(Refer Slide Time: 30:08)




So, I will write that down also. Hence, we can, hence if we pass message along the edges of the Tanner graph in exactly the same manner as in the case of, of messages passed along the edges of, of the junction tree associated with the example 7 4 2 block code, then when edge neighborhoods are tree like, the messages passed along the edges of the Tanner graph can, can be interpreted as conditional beliefs. So, that was the point that I want to make. So, this, so this message passing, as in the case of the junction tree, is called belief propagation base message passing. So, from now on we will assume that messages are passed accordingly.

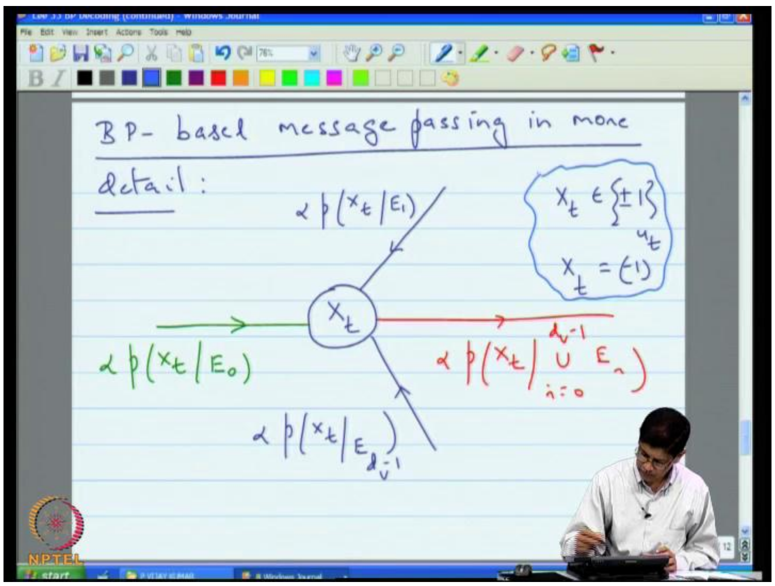
(Refer Slide Time: 32:10)




So, then, now let us look at a message in general that is passed at a check node. So, so, belief propagation based message passing in more detail. At a variable node, at a variable node, let us say, corresponding to x t, then you would first of all have an incoming local canal, which should actually give you p of x t given y t. Now, let me just rewrite that slightly differently. I want to actually say, that I want to view all the conditions as being representing some evidence and I will explain that in slightly more detail shortly. Then, you have any incoming messages here, which you can write as p of x t given E 1, p of x t given E d v minus 1 and then, you have an outgoing message, which is then, p of x t given the union i equal to 0 to d v minus 1 E i. And all the messages are actually, this should be a proportionality constant included here, alright. So, hopefully from the discussion that I just told you, this part is actually clear, that is, every message that is being passed is actually because we saw that here, right. I mean, because we saw that.

Let us take, for example, here the message that was passed here was an indication about the belief of x 4 based on this, right. At a different junction, here again it was about the belief of x 4 is based on more evidence. So, the evidence, here in this particular case, for instance, was based on the received symbol y 3, y 3, y 4, y 5, y 6, y 7, as well as, these two parity checks. But though, that evidence is really the union of two pieces of evidence, one coming from the knowledge of y 3, y 6 and the parity b check b and the other coming from knowledge of y, y 5, y 6 and the parity actually, see the union of three pieces of evidences. Because the collecting evidences from this segment here, from this segment here, as well as, the local information here collecting information. And so, what you actually, put out then here is the conditional probability given the union of the evidence. So it is, is in this sense, that I am actually saying, that is what happens in general.

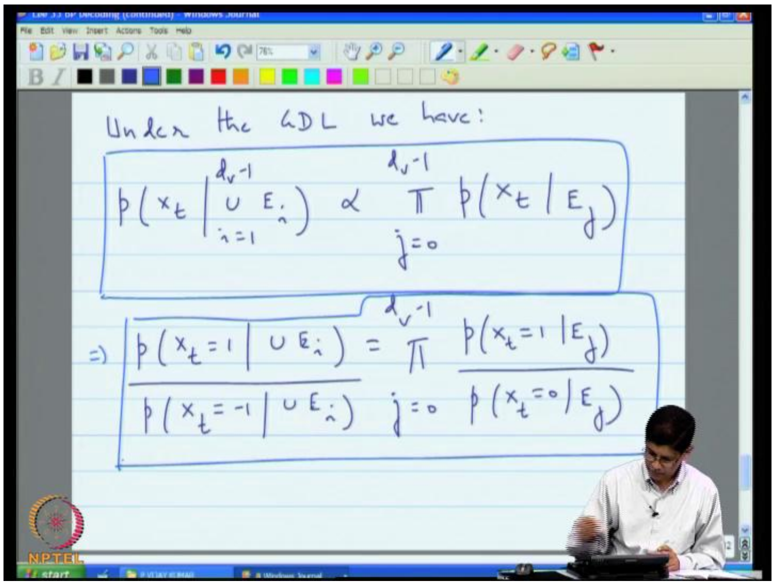
(Refer Slide Time: 35:50)




So, in general you are passing messages like this and what you have to do to this? Then, actual message then is proportional to these quantities, but you passed messages just the way you do on the junction tree, you would take the incoming messages, you take a local canal and you form the product. Remember, that the messages that you are passing is really the vector because this function you have to pass. This function means, you have to pass this quaintly as the function of x t since x t can take on value 0 or 1.

We are actually passing on vector with two components, so that is what was happening here. However, we want to, now since the two of the vectors that you are passing are probabilities and their sum is 1. It is reasonable to assume, that you could get away by just passing a single probability instead of a pair of probability, so it turns out, that you can do that and in, you can end up just passing likelihood ratio. Let me show how that actually works.

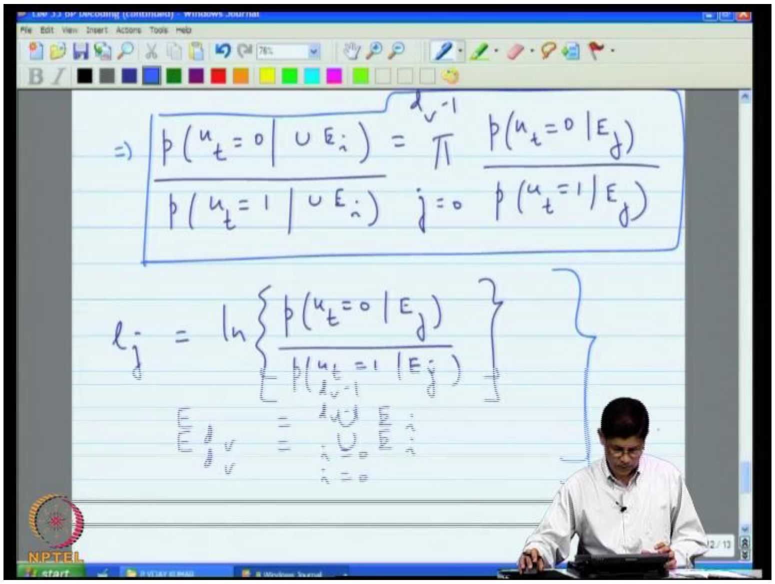
(Refer Slide Time: 36:53)




So, under the GDL we have, that p of x t union I is equal to 1 to d v minus 1 of E i is proportional to the product j is equal to 0 to d v menus 1 p of x t given E j. Now, let me rewrite this by spelling this out for x t equal to 0 and 1. This, this whole is for both x t equal to 0 and 1. If I take the ratio, if I write this for x t equal to 0 and then I write for x t equal to 1 and divide both sides, then the proportionality sign will go away and I will have equality. So, I can write, so this implies, that the probability now, so I have to be a little bit careful here.

Now, I can either interpret, so here I guess, in accordance with earlier notations these x t's are really belonging to the set plus minus 1, let me write x t equal to minus 1 to the u t. So, when u t is either 0 or 1, so I can instead write this as the probability of, so I can say, that probability of x t equals 1 given the union of the E i divided by the probability of x t equal to minus 1 given the union of the E i is equal to the product j is equal to 0 to d v minus 1 probability x t equal to 1 given E j divided by the probability x t is equal to 0 given E j. So, this is what you would have.

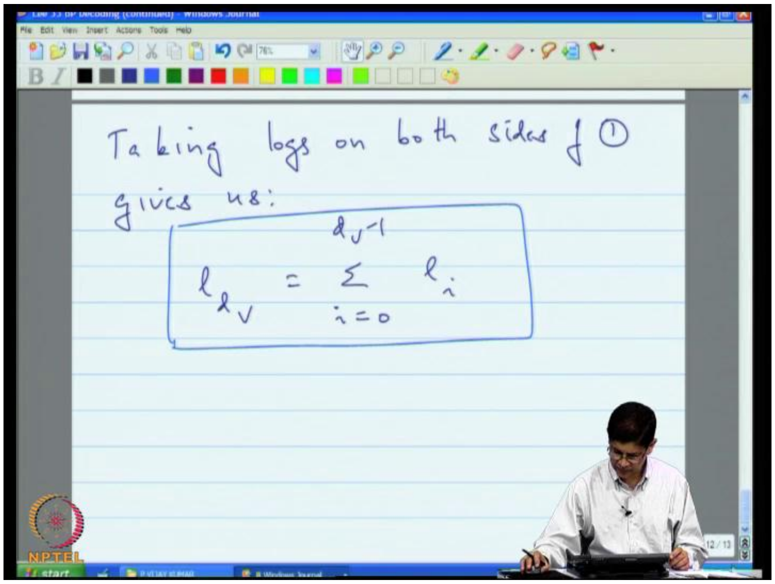
(Refer Slide Time: 40:12)



And in terms of, if I switch this and want to rewrite this in term of not x j's, but rather the u j's, so I would replace these symbols by u. This should have been minus 1, excuse me. So, I can write here minus 1, so I can write, so I can write this instead as u t equal to 0, u t equal to 1, u t equal to 0, u t equal to 1. So, I switched from u t in terms of u t, rather than x t. So, what this is telling you is, that here, whereas here we were passing function where each function is actually a vector here, what we are doing is we are passing a ratio, which is the scalar.

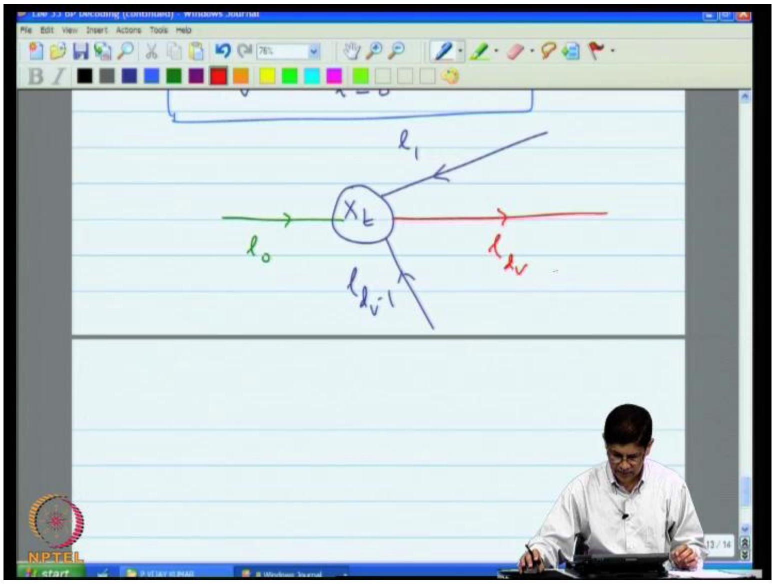
And now, the proportionality sign has also gone away and what we will actually do is, we will, this, the log likelihood, this is the likelihood ratio, we will actually pass on to the log likelihood ratio. Let us define, let us define l j to be l n of p of u t equal to 0 given E j divided by u t equal to 1 given E j. And let us also denote, let us denote by E d v, the union of the evidence. So, let us take these two definitions then.

(Refer Slide Time: 42:46)




If I call this equation 1, then in the log domain, taking logs on both sides of 1 gives us, that you have l d c is the sum i is equal to 0 to d v, l d v sorry is the sum l i, i is equal to 0 to d v minus 1. So, just want to reiterate, that our message passing algorithm has, in a sense, remained the same because what we were doing under the junction tree was we were passing a function, where the two quantities are proportional to probabilities. But you can actually make do because if your interest is only, ultimately our interest is only or rather, we can recover the two probabilities even if we have access to only the ratio because after all we know that the sum of the probabilities is 1. So, you can make do by sending a single real number as opposed to a pair of real numbers.

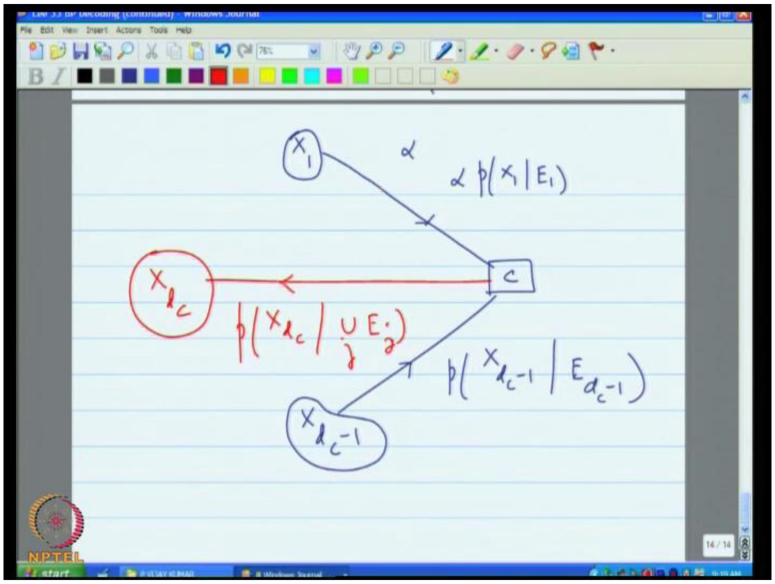
(Refer Slide Time: 44:40)




There is a slight increase in complexity of the operation that is carried out at the check node. At the variable node things, that has even become simpler because now you have a view, that rather than passing these conditional beliefs, which we had here, when I am going to have the same diagram, that except we are actually going to pass, we are just going to pass likelihood ratio. (Refer Slide Time: 44:54)

So, let me just draw such a graph, so you have your node, which I will still label as x t and now, I will say, that what is incoming is the likelihood ratio, which is l 1. What is coming from the other nodes also, are also likelihood ratios based on the same code symbol except that the evidence on which they are passed is different. And then, what you are actually computing is l d v and we are computing this according to this . So, this is the message passing that is carried out at a variable node.

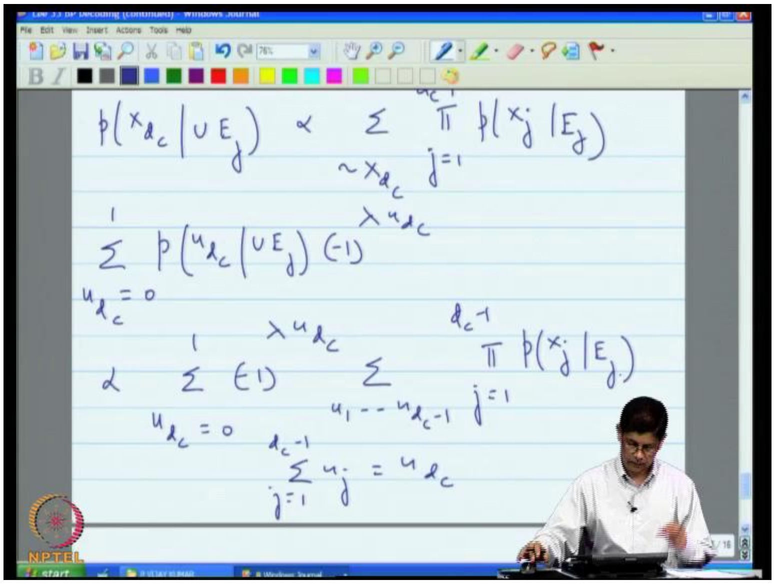
(Refer Slide Time: 45:58)




Now, what happens at a check node? At a check node things are little bit more complicated. In terms of function there is no problem because we, we know, that if you have, let us say, a check node, a check node C, then what is going to be. You are going to have, when we are passing beliefs earlier and the belief messages were quantity, so this would be proposals. Let us say, that the other end of this, there was the variable x 1 and let us say, that here this correspond to x d c minus 1 and let us say, that this corresponds to x d c and so, these respectably belief. So, this belief would actually be proportional to p of x 1 given some evidence; this would be proportional to p of x d c minus 1 given some evidence. And what the message, that is actually going out would then be p of x d c given the union of this evidence.

So, now how do we in a way, instance, it is proportional to… So, the question is how do you actually represent this? I should add few dots  because you could have some other notes as well. So, how do you represent this in terms of the log likelihood domain?

(Refer Slide Time: 47:59)

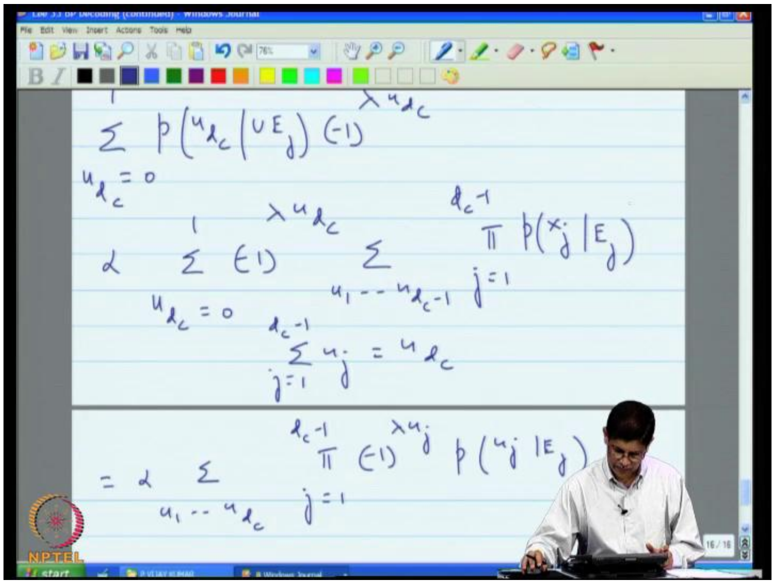



Now, we know, that if you were to write down an expression, we know, that p of x d c given the union of the evidence is proportional to the, to the sum over x d c of the product j is equal to 1 d c minus 1 p of x j given E j. So, that means, that here what is happening, that you are doing message passing, you are taking all the incoming messages, then you are marginalizing. Now, this parity check would typically be associated under the GDL to the following set of variables. You would have x 1, x 2, x d c. So, you might write this node in this format.

Now, when you see, that you will see, that the outgoing message must marginalize with respect to all the other variables except x d c, so that is what this not sum represents. It means, that you are marginalizing or summing over all possible values of the remaining variables, alright. You can turn it into an expression using in, in involving likelihood ratio as follows. This is a little bit of trick, so we will sum, we will take this and sum over u d c equal to 0 to 1. Now, I am going to switch to the u's rather than the x's.

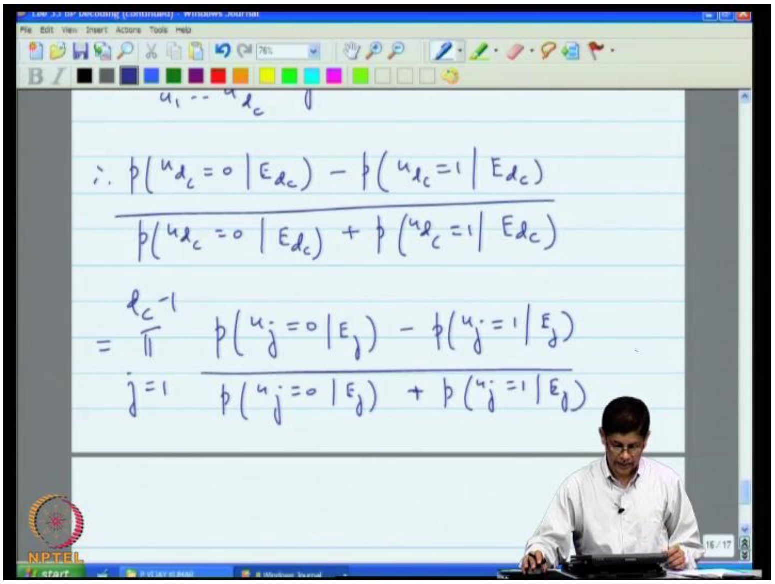
So, I am going to form this sum for the reasons, that will become clear and this is going to be proportional to the sum u d c is equal to 0 to 1 minus 1 to the lambda u d c the subject or the sum u 1 through u dc minus 1 subject to sigma u j is equal to the u d c. And here, you would have, should have this expression and now, just an observation. So, you are summing over u d c and over all the other variables, but subject to this condition.

(Refer Slide Time: 51:01)



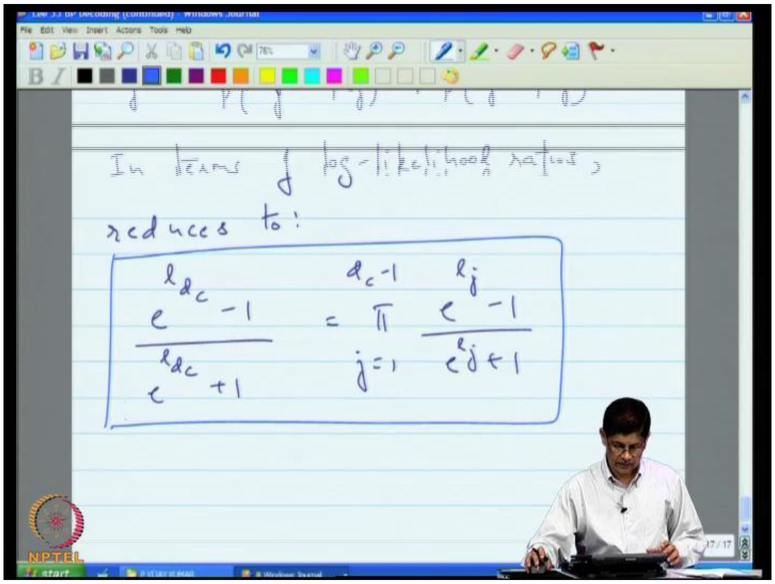
So, that means, that you can actually replace. If you replace the u d c term here by the sum of these variables, then you can actually get rid of the sum over u d c. So, this can be rewritten as being proportional to the sum over u 1 through u d c of the product j equal to 1 to d c minus 1 minus 1 to the lambda u j p of u j given E j . So, this should, pardon me, I, so this should have been u j. Remember, that x j is minus 1 to the u j. So, there is no difference between belief about x j and the corresponding belief about, and the belief about corresponding value of u j. So, it turns out to be more convenient to deal with the u's rather than the x's. So, that is what I have actually done here.

(Refer Slide Time: 52:43)




So, now I am going to actually have a sum here, I am going to take the ratio of this quantity with lambda equal to, lambda equal to 1 in the numerator and lambda equal to 0 in the denominator. So, what that will actually give me is that, therefore, therefore p of u d c equal to 0 given E d c minus p of u d c equal to 1 given E d c divided by the sum p of u d c equal to 0 plus p of u d c equal to 1 given e d c is, then the product j is equal to 1 to d c minus 1. So, I have this sum and now you can see the connection with log likelihood ratios because this can be rewritten in the term of log likelihood ratios, of likelihood ratio because I am just going to divide. Let us say, I divided this pi p of u d c equal to 1, so I will get the likelihood ratio here minus 1 and similarly, over here.

(Refer Slide Time: 54:32)




So, this can be, this again, in terms of log likelihood ratios reduces to e to the l d c minus 1 upon e to the l d c pulse 1 is the product. j is equal to 1 to d c minus 1 e to the l j minus 1 upon e to the l j plus. So, you get an expression like this and we just have about a minute to two minutes left, so let me just summarize.

So, what I tried to actually do was, I was trying to explain to you how belief propagation message passing works on the Tanner graph and I said really something, that already you know because you know it in the setting of the junction tree and what we do is, we just port the algorithm from the junction tree to the Tanner graph, and everything stays the same as long as your neighborhood is tree like and we saw that. So, you have an interpretation of the messages as beliefs and now, because they are beliefs it turns out, that instead of passing messages you can actually pass log likelihood ratios. We wrote down the expression for the message, that is passed across the variable node in term of log likelihood ratio and we are just about to do that for the check node. But since we are running out of time, let us continue this; we will continue this part of the discussion in our next class.

So, thank you.