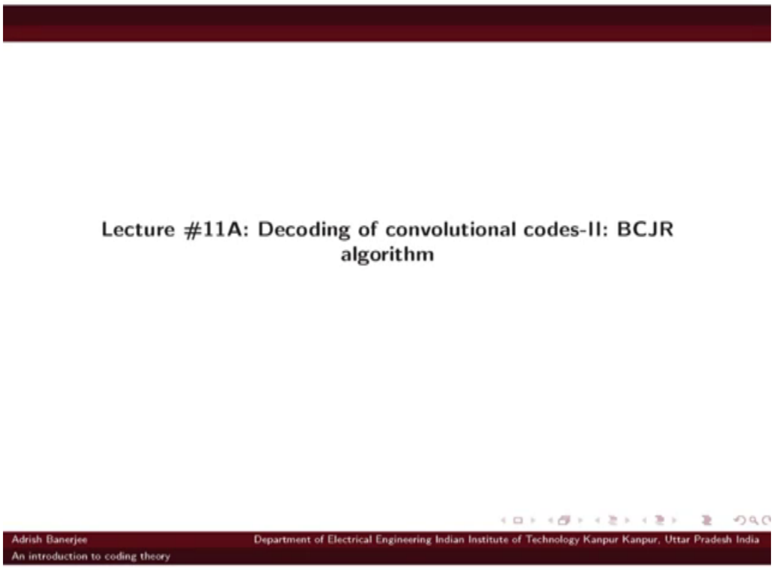
An Introduction to Coding Theory
Professor Adrish Banerji
Department of Electrical Engineering
Indian Institute of Technology, Kanpur
Module 05
Lecture Number 20
Decoding of convolutional codes-II BCJR algorithm

(Refer Slide Time 00:14)

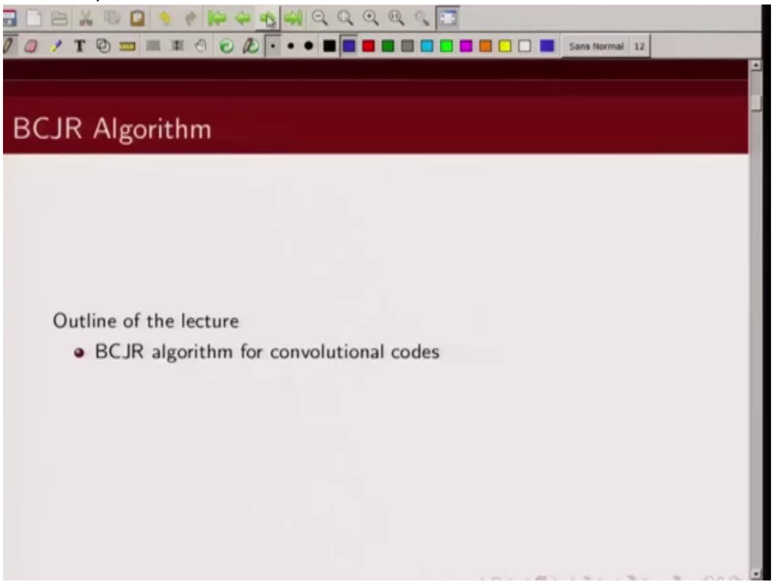


We are going to continue our discussion on decoding of convolutional codes. In the last class

(Refer Slide Time 00:20)

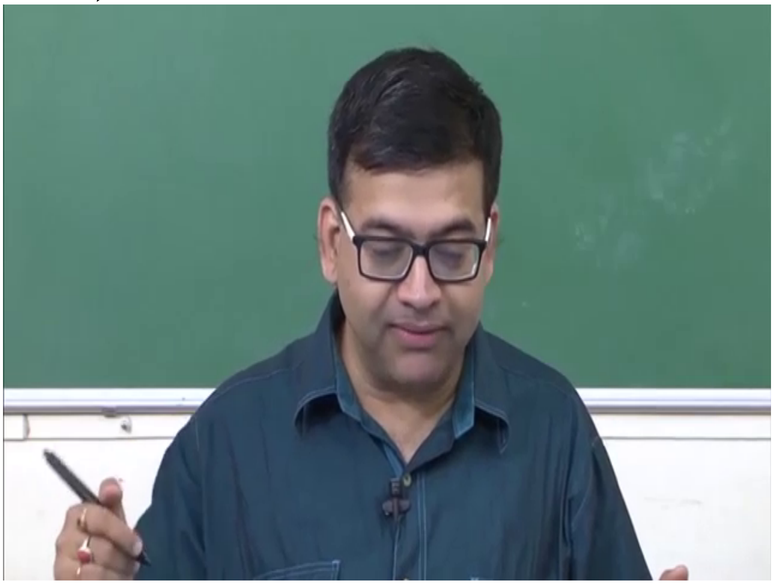
we talked about Viterbi decoding. And if you recall Viterbi decoding is an efficient algorithm to compute a path to the Trellis of a convolutional code. Now it essentially finds out, Viterbi algorithm essentially finds out an estimate of the codeword because any path through the Trellis of a convolutional code is basically a codeword. Now that not necessarily minimizes the bit error rate probability. In many applications we are interested to minimize the bit error rate. So

(Refer Slide Time 01:01)



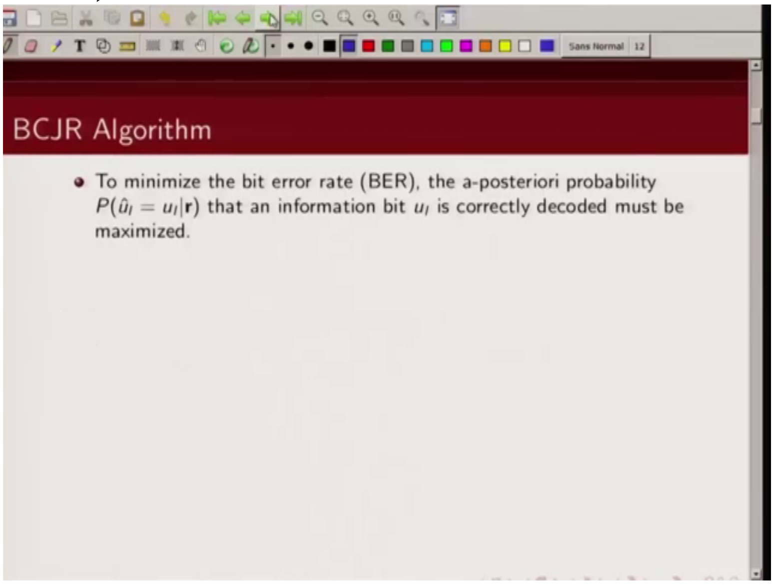
today we are going to talk about a decoding algorithm which is basically going to minimize

(Refer Slide Time 01:09)



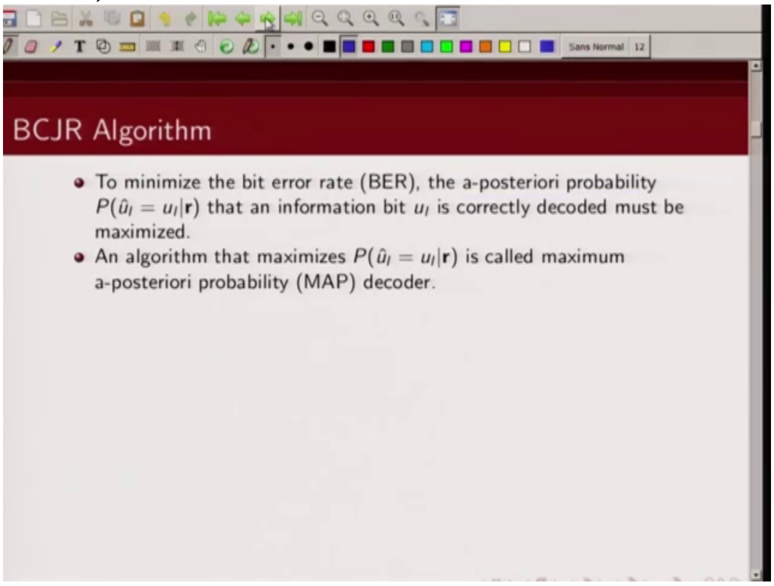
bit error rate probability, symbol error rate probability.

(Refer Slide Time 01:12)



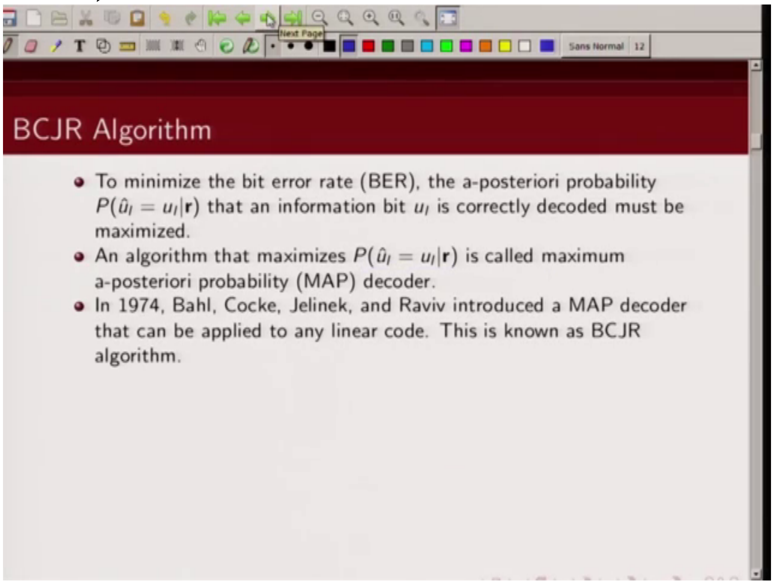
So we are going to use a posteriori probability based algorithm to estimate our information sequence. And this

(Refer Slide Time 01:25)



algorithm which maximizes probability of u hat given u given the recieved sequence r is known as MAP decoder. Now this is known as, also known as

(Refer Slide Time 01:41)

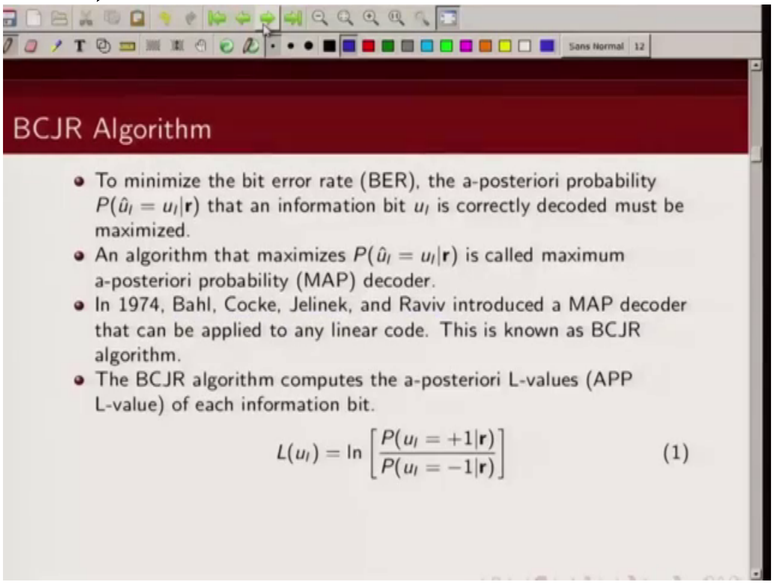


B C J R algorithm named after these researchers who, Bahl, Cocke, Jelinek and Raviv, who introduced this algorithm in 1974. And this algorithm can be applied to any linear code, block code or

(Refer Slide Time 01:58)

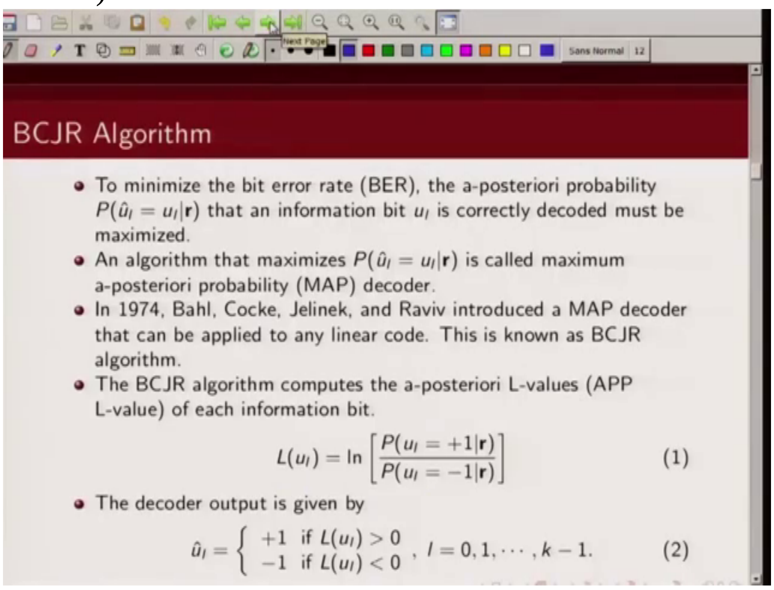
convolutional code.

(Refer Slide Time 02:02)



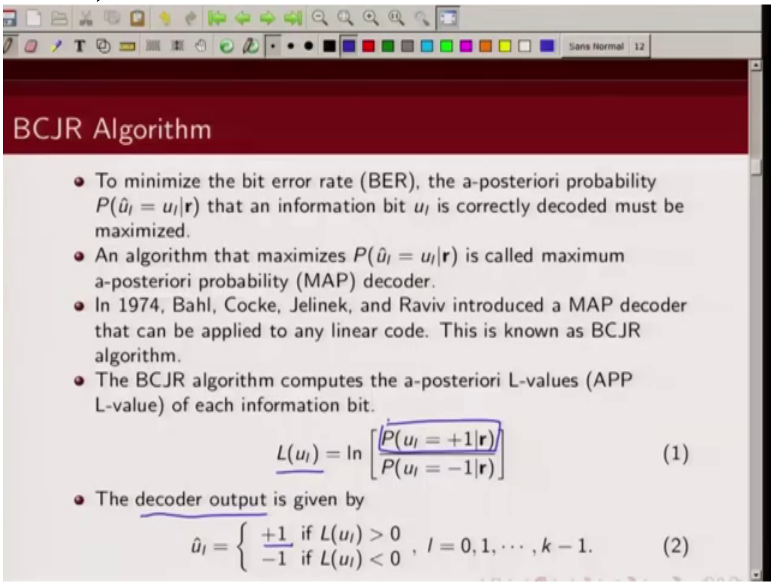
Now the complexity of this algorithm was much higher than Viterbi algorithm and that's why it was not popular in 70s, but in late 90s when, this concatenated codes, turbo codes came into picture and we required soft estimates then these algorithms became very, very popular. So what this algorithm does, it computes the a posteriori probability. So I define a posteriori, Log-likelihood value, I call it L value like this. So it basically computes probability of u l being plus 1 given a received sequence r by probability of u l being minus 1 given recieved sequence r. Take a log of that. Now if this L value is greater than zero, then you decide in favor of u l being plus 1, otherwise you decide in favor of u l being minus 1.

(Refer Slide Time 03:10)



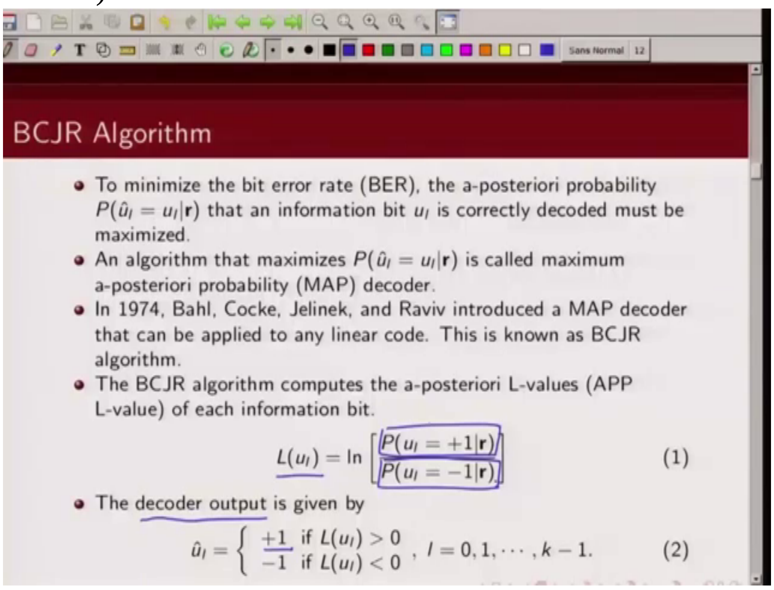
So your decoder output will be plus 1 if the L value is greater than 0, otherwise you decide in favor of minus 1. So we are now going to talk about how to compute these terms, these terms you see in computation of

(Refer Slide Time 03:31)



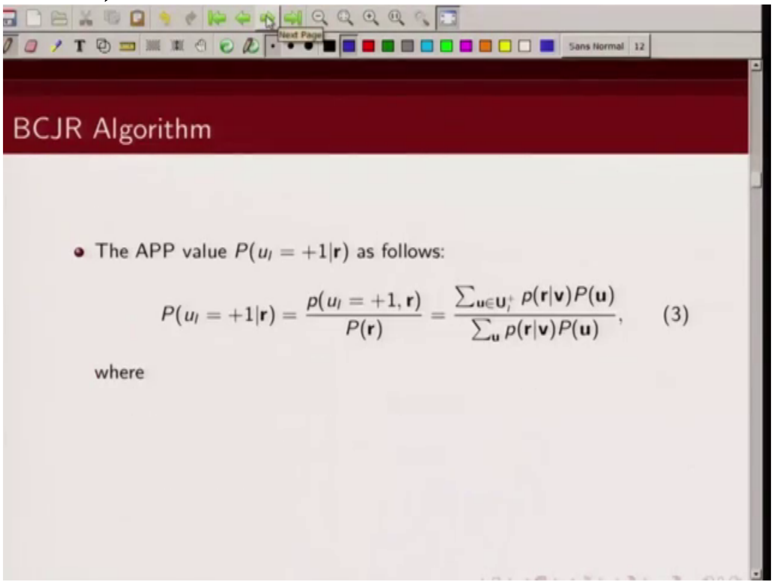
A P P value, how do we compute these terms and how we can

(Refer Slide Time 03:37)



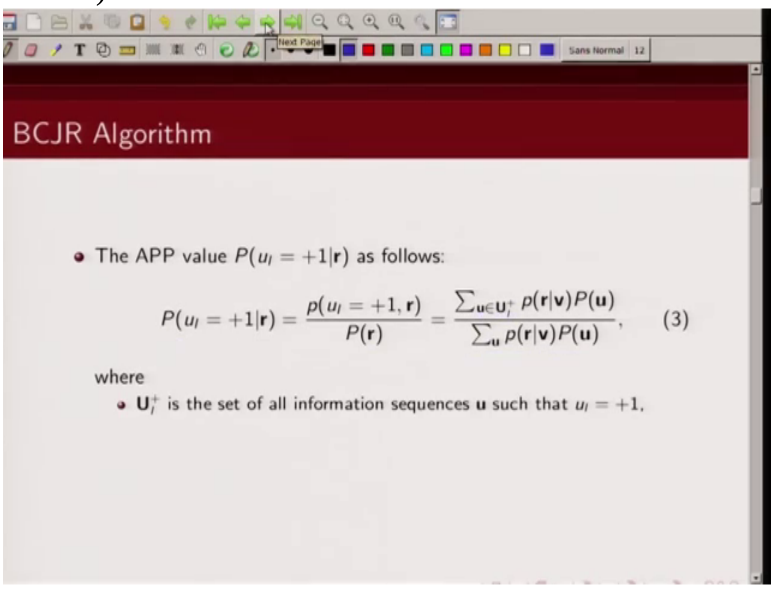
exploit the structure of the Trellis of the convolutional encoder to simplify this expression.

(Refer Slide Time 03:46)



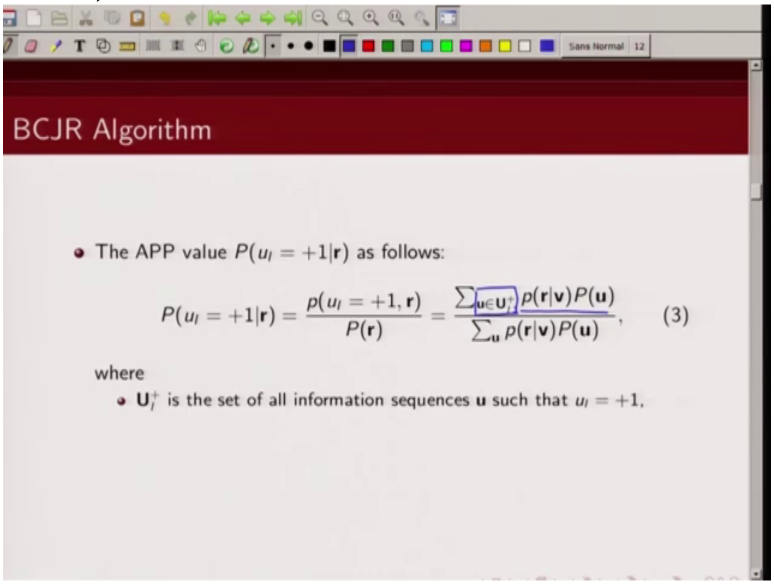
So let us look at this probability of u l being plus 1 given a received sequence r, this can be written as joint probability of u l being plus 1 and recieved sequence r divided by probability of receiving this r. Now this probability of u l being plus 1 given a recieved sequence r can be written as probability of r given v multiplied by probability of u sum over all input sequences that belongs to the set where u l is plus 1 and this can be written as probability of r given v multiplied by probability of u sum over all input sequences.

(Refer Slide Time 04:34)



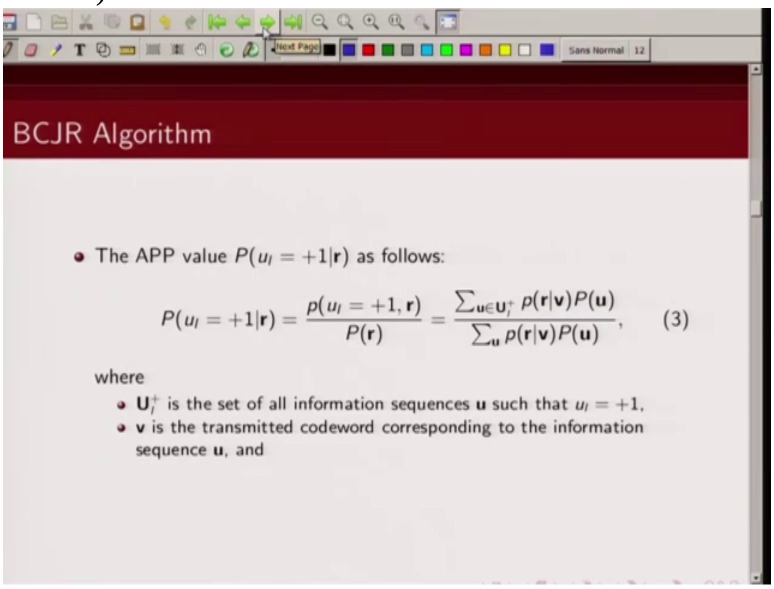
So as I said, since we are interested in joint probability of u l being plus 1 and r we sum this probability over all those set of

(Refer Slide Time 04:47)



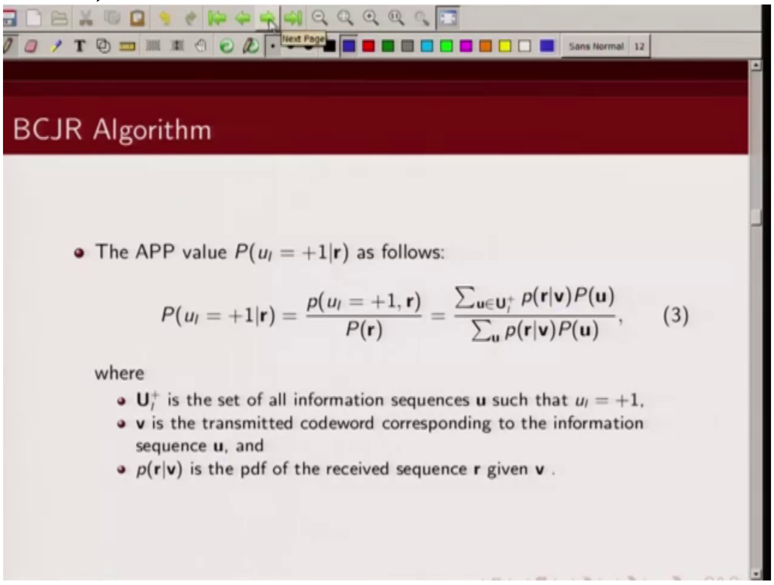
information sequences where the bit, the corresponding bit is plus 1.

(Refer Slide Time 04:57)



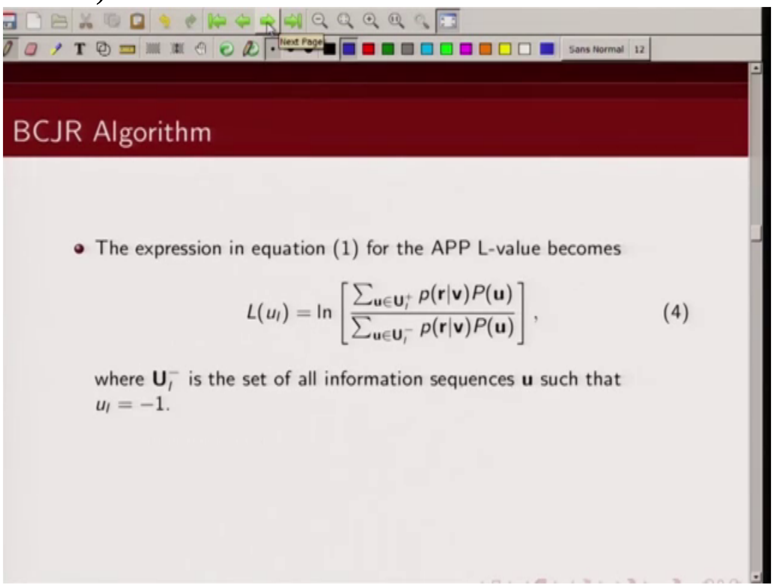
And our transmitted codeword is v, our information sequence is u and r is the recieved sequence. Probability

(Refer Slide Time 05:07)



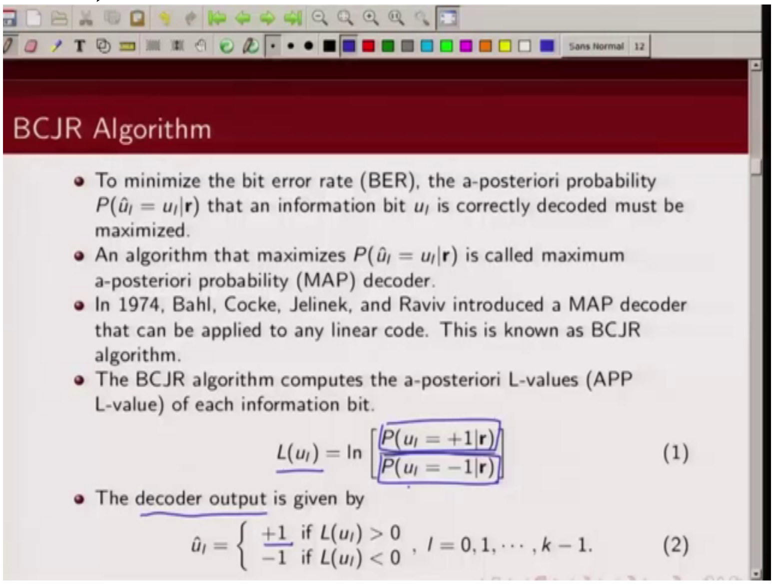
of r given v can be computed from the channel, given channel.

(Refer Slide Time 05:14)



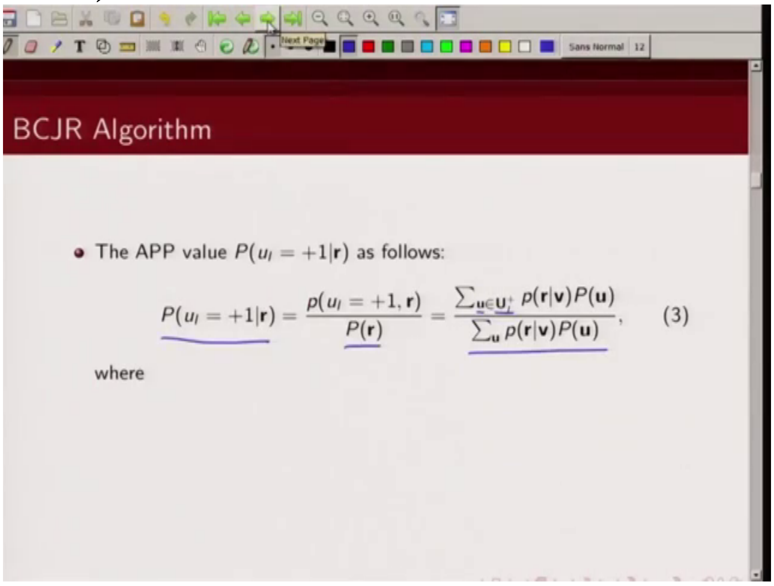
Similarly we can also compute, now if you go back here, the denominator we need to compute probability of

(Refer Slide Time 05:24)



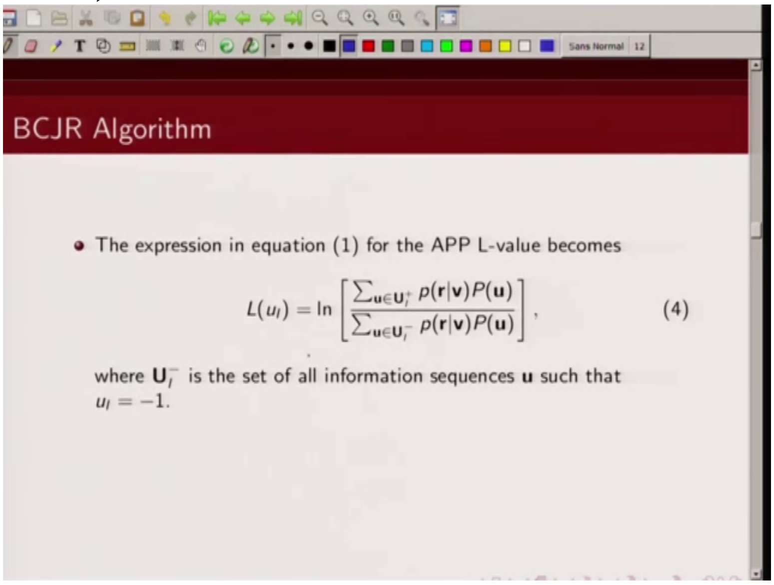
u l being minus 1 given r so similar to this term we can also write probability of u l being

(Refer Slide Time 05:32)



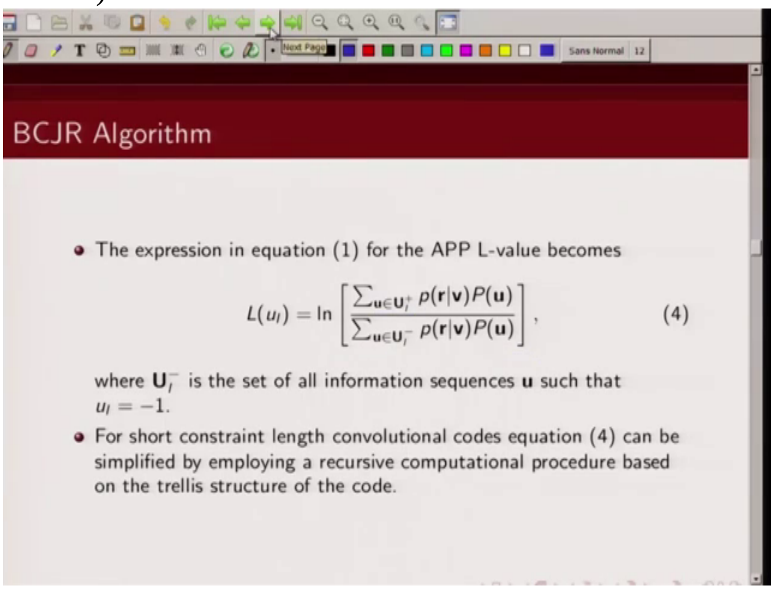
minus 1 given r. And probability of r is a common term. So if we do that,

(Refer Slide Time 05:40)



what we get is this. So again this L value, the A P P value of u of l is given by probability of r given v multiplied by probability of u where we are summing over all information sequences where the corresponding bit is plus 1. And similarly for the denominator we are summing over all information sequences where information bit is minus 1. We will illustrate this with the help of example and then things will be little more clear.

(Refer Slide Time 06:26)

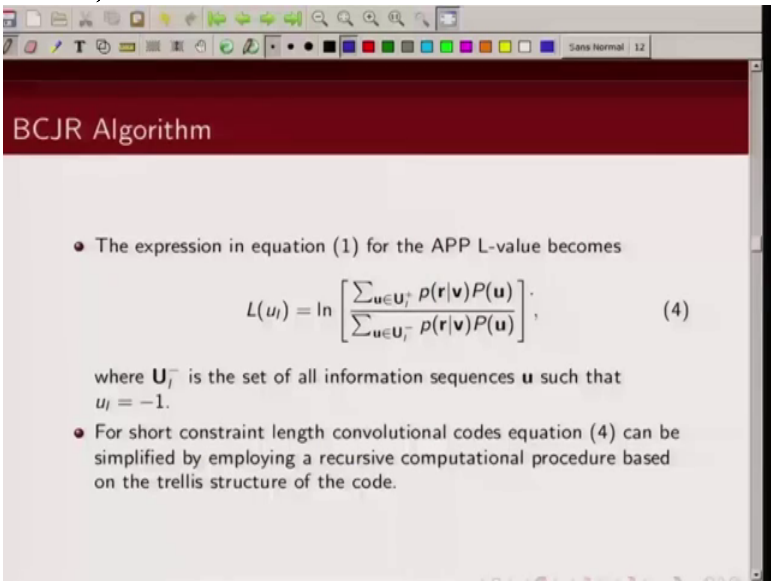


Now note here, if you have very large sequences, this is sum over all input sequences where u l is plus 1 and this is sum over all input sequences where u l is minus 1. So if your information sequence is large this is sum over very large number of possibilities. So this is quite complex.

(Refer Slide Time 06:49)

Now can we use the structure of the convolutional code to simplify this expression?

(Refer Slide Time 06:56)

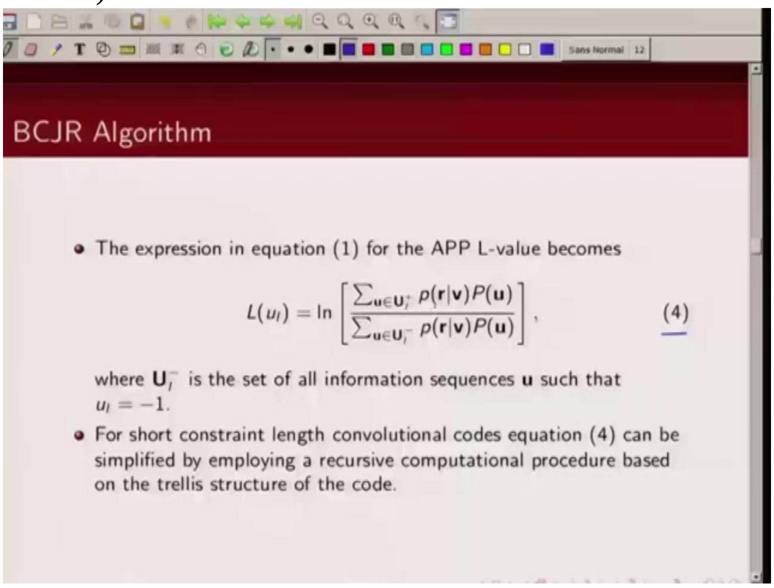


The answer to this is yes. So we are going to basically simplify this equation 4 by using the Trellis structure of

(Refer Slide Time 07:08)

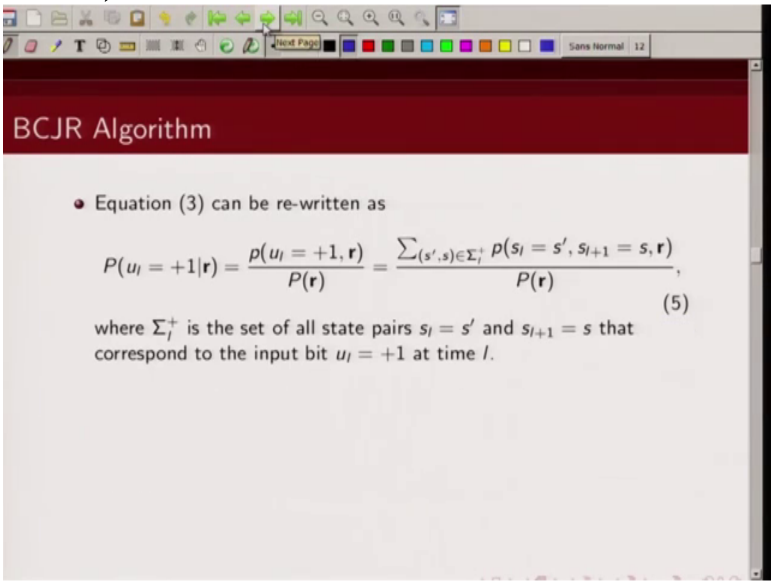
the convolutional code. We know all possible transitions are not possible. So our Trellis diagram or the state diagram will, will ensure, will tell us what are the valid transitions. So we can simplify

(Refer Slide Time 07:22)



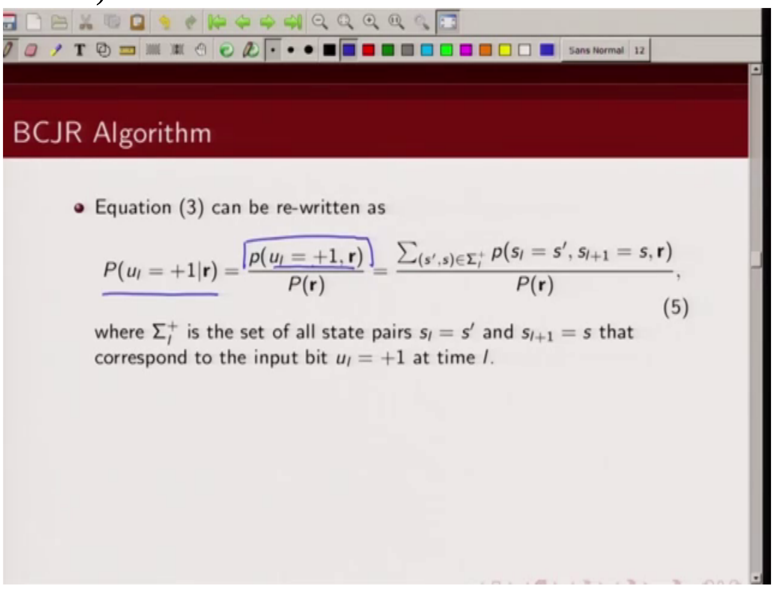
this expression using our valid state transitions. So what we are going to do is we are going to make use of the Trellis structure of the code to simplify our equation number 4.

(Refer Slide Time 07:42)



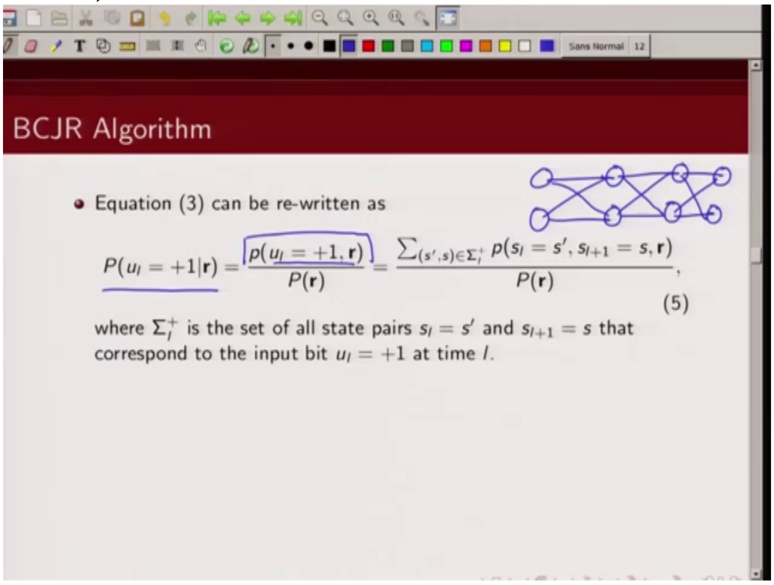
So let us see how do we do it. We again go back and look at this probability of this u l being plus 1 given our received sequence r as we have written, this can be written as joint probability of u l being 1 and the probability of receiving r divided by probability of r. Now we are going to, now look at this expression. This is joint probability of u l being plus 1 and

(Refer Slide Time 08:13)



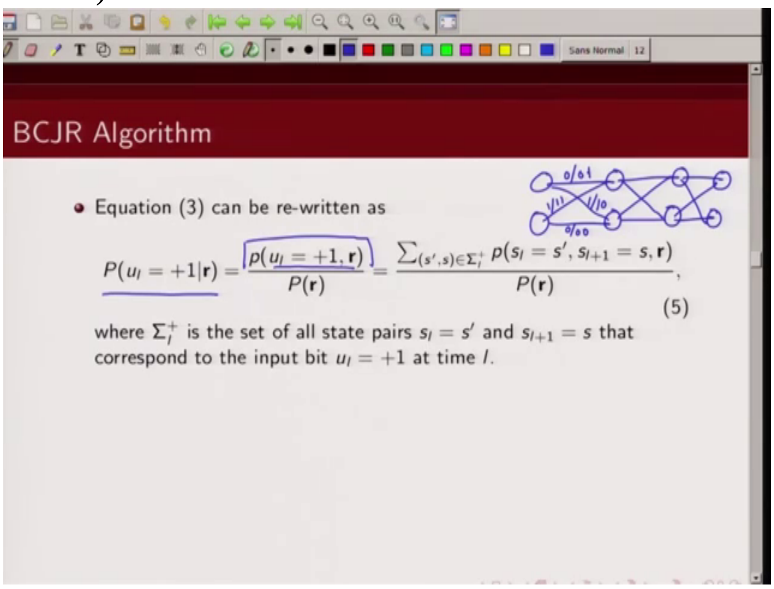
given the recieved sequence r has been recieved. So if you look at any Trellis diagram, let's say this is some Trellis diagram, simple 2 state code, like that you have, so we are interested in where u l is

(Refer Slide Time 08:38)



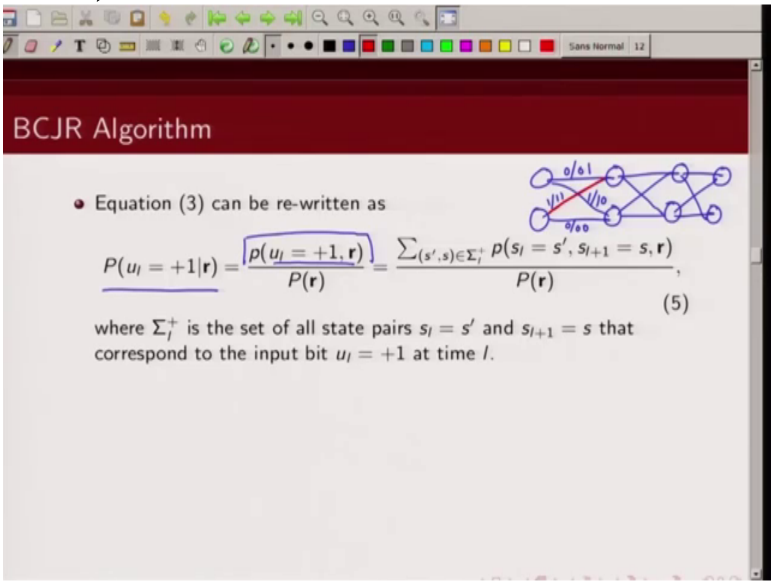
plus 1 and where u l is minus 1. Let us say this is 0 by 0 0, this is 1 by 1 1, this is, let's say 1 by 1 0, this is 0 by 0 1.

(Refer Slide Time 08:52)



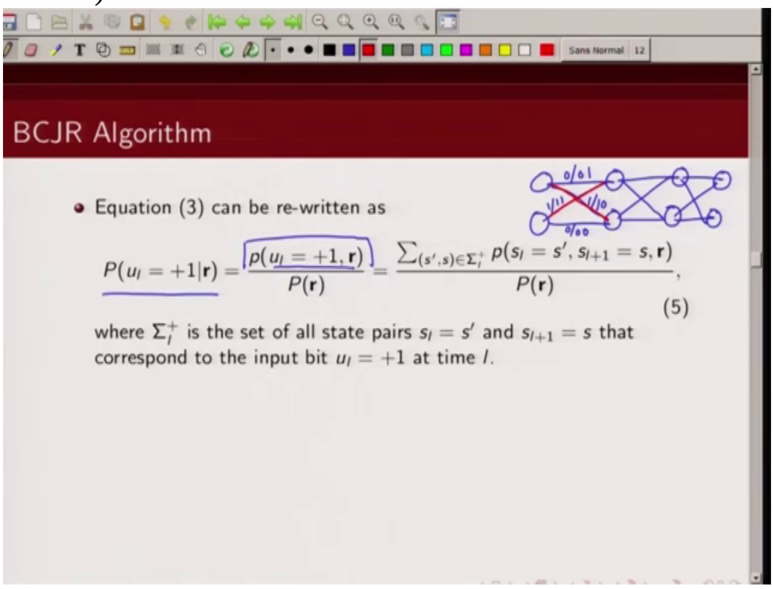
So let's look at one Trellis section. So we are interested in all those transitions which belongs to u l plus 1. Now what are those transitions? So in this example this is one such

(Refer Slide Time 09:07)



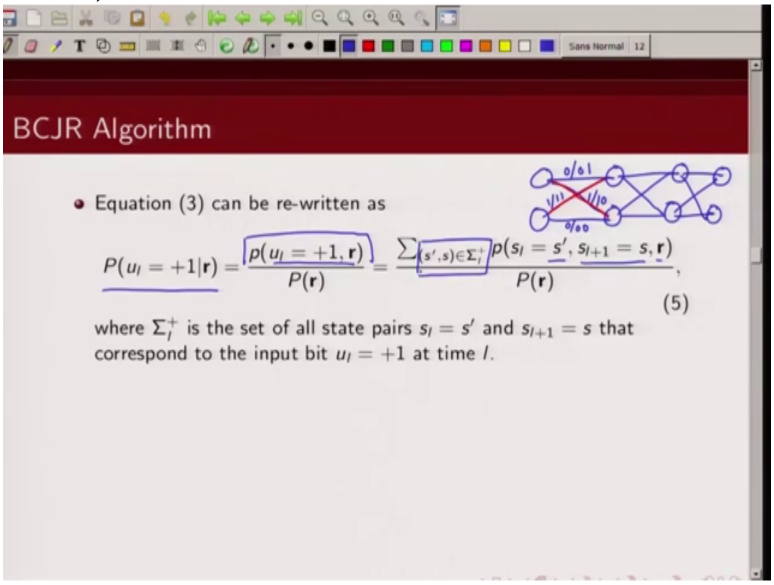
transition. And the other is this transition,

(Refer Slide Time 09:11)



Ok. So what I am writing here is then I am interested in what's the joint probability that the previous state is s prime, the next state is s and the recieved sequence is r and I am summing over all those

(Refer Slide Time 09:35)

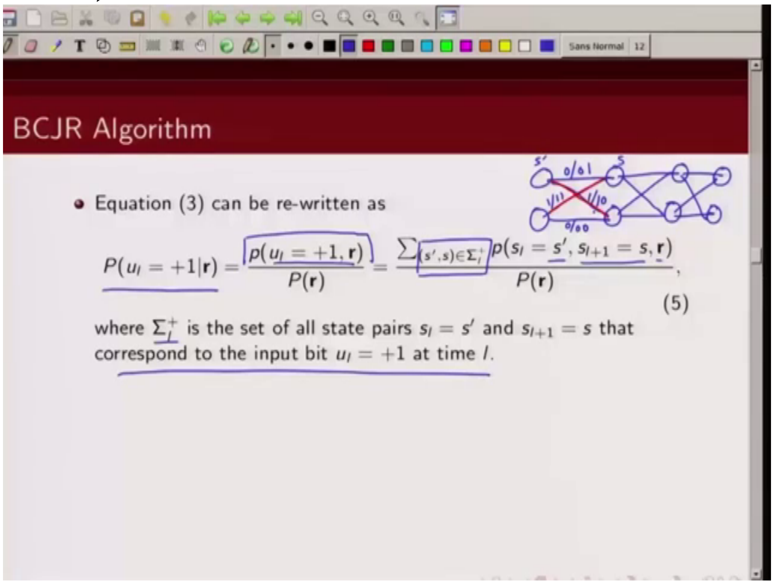


state transitions that belong to the set pair where the input corresponds to this transition is plus 1. So note what is my this sigma l plus, it is a set of all state pairs where the initial state is s prime then next state is s so its, it's a pair of states where the transitions, the input bit corresponding to a valid transition is plus 1. So, so in this case the set that belongs to this is given by this red line, Ok.

(Refer Slide Time 10:24)

So I can write the joint probability of u l being plus 1 and r in terms

(Refer Slide Time 10:31)

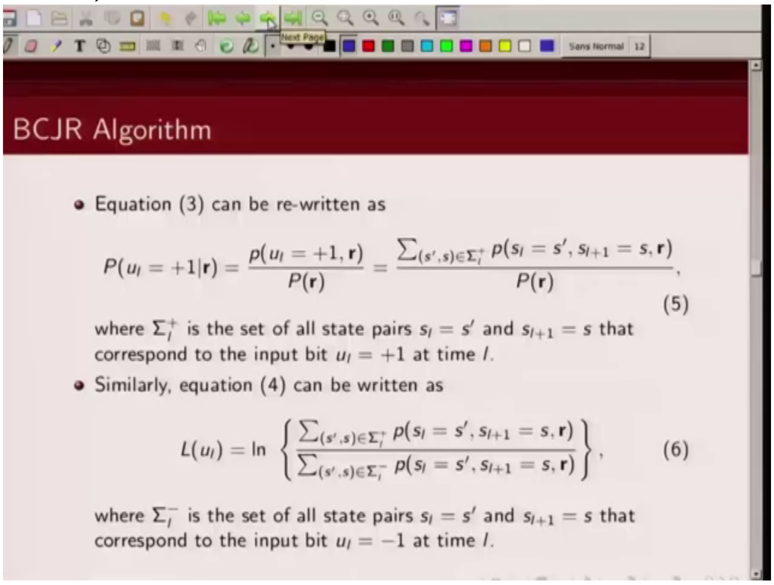


of condition on the valid Trellis transitions in this way, I can write it as what is the probability that the initial state is s prime, next state is s given the received sequence r and I sum over all those transitions which belong to input bit being plus 1.

(Refer Slide Time 10:56)

Similarly

(Refer Slide Time 11:00)

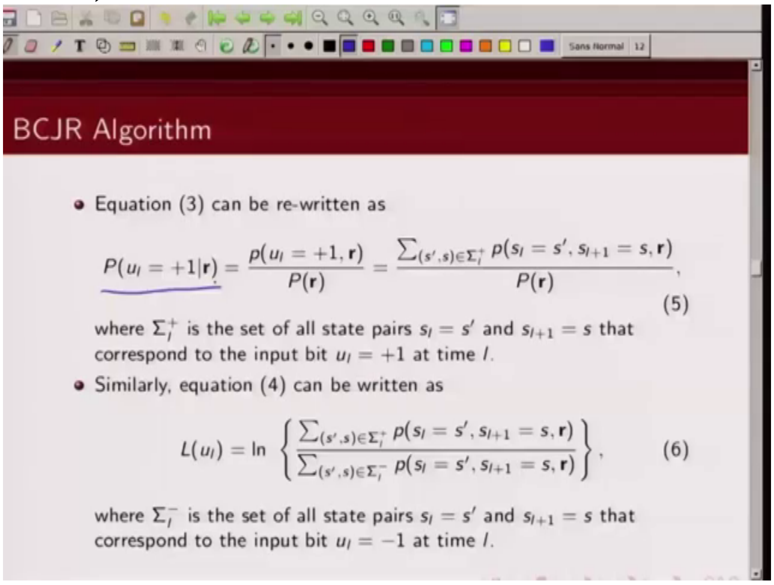


I can write exactly the way I wrote, probability of u l being plus 1 given r, I can follow the same procedure to write what is the probability of u l being minus 1 given r. So what would be the change here? So I will compute this probability and I will sum over all those state pairs which correspond to

(Refer Slide Time 11:25)

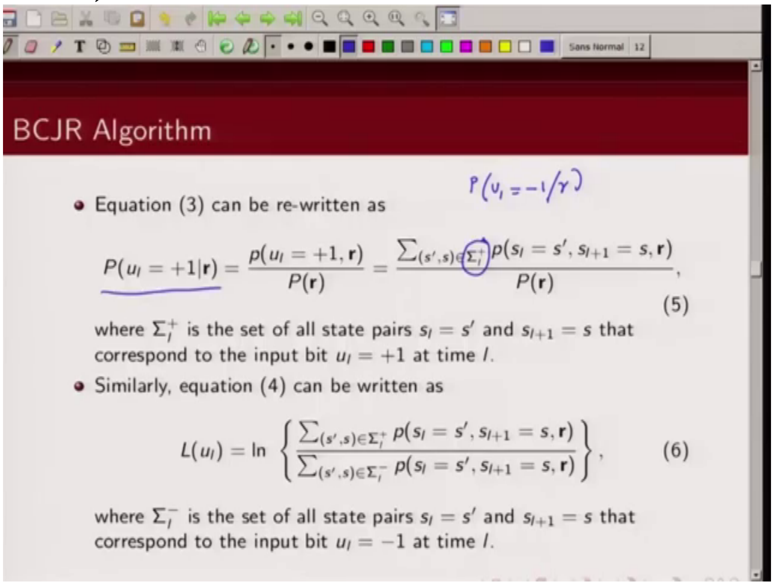
input bit minus 1. So if I plug

(Refer Slide Time 11:30)



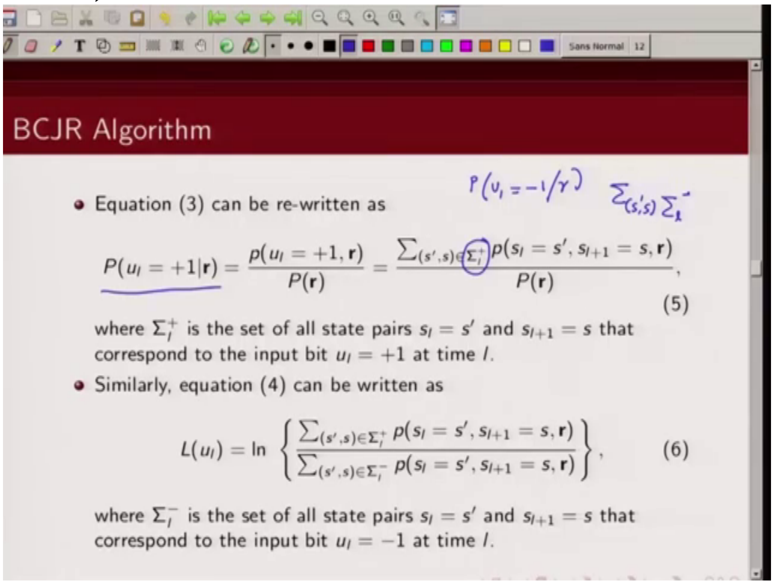
these values of probabilities which are given by equation 5 and similarly I can calculate the probability of u l being minus 1 given r so instead of this thing here

(Refer Slide Time 11:45)



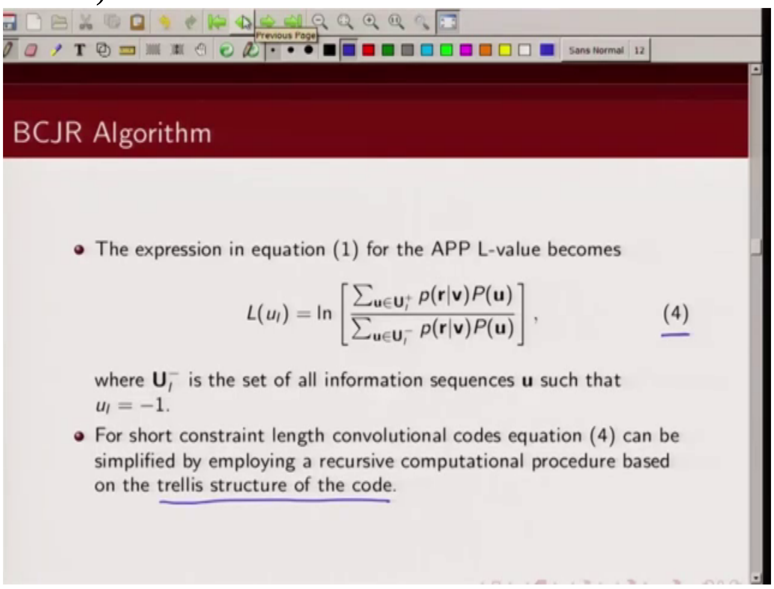
I will have summation over s prime as summation over all those pairs which corresponds to u l being minus 1

(Refer Slide Time 11:54)



and I will get this same thing here. So if I do that what I will get is equation number 6. So note that previously I had the same expression,equation number 4

(Refer Slide Time 12:10)

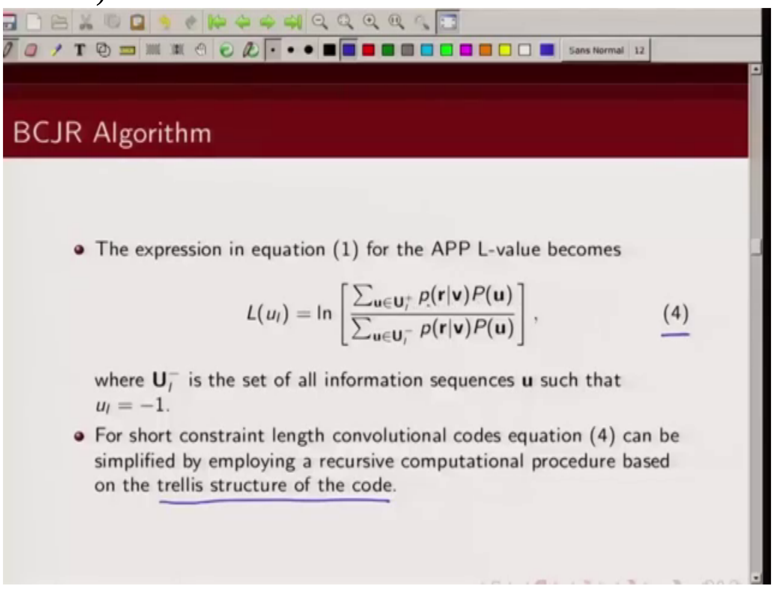


in terms of this input sequence u l. Now if our

(Refer Slide Time 12:17)

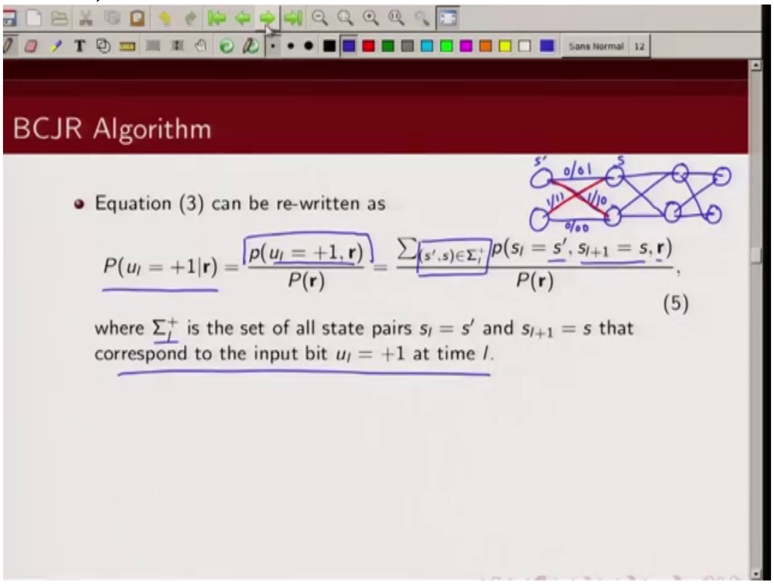
input sequence is very long this is summation over a large number of

(Refer Slide Time 12:20)



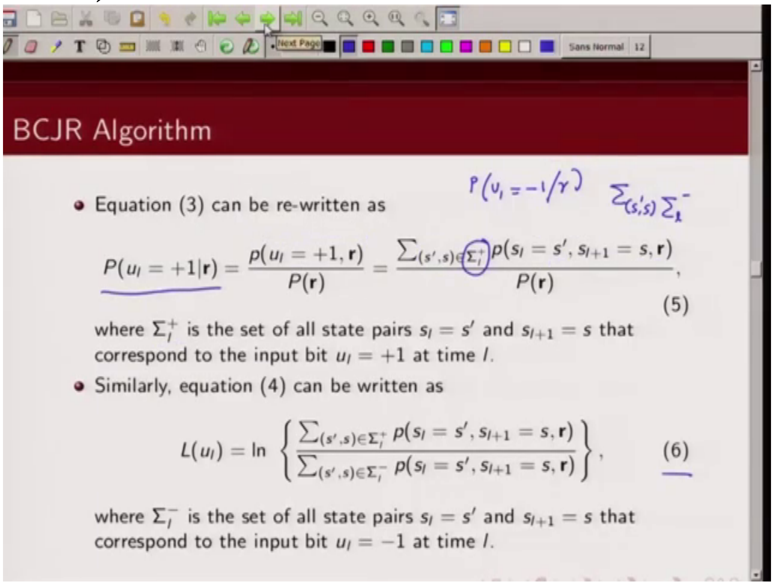
terms where as I have now

(Refer Slide Time 12:23)



simplified my expression.

(Refer Slide Time 12:25)

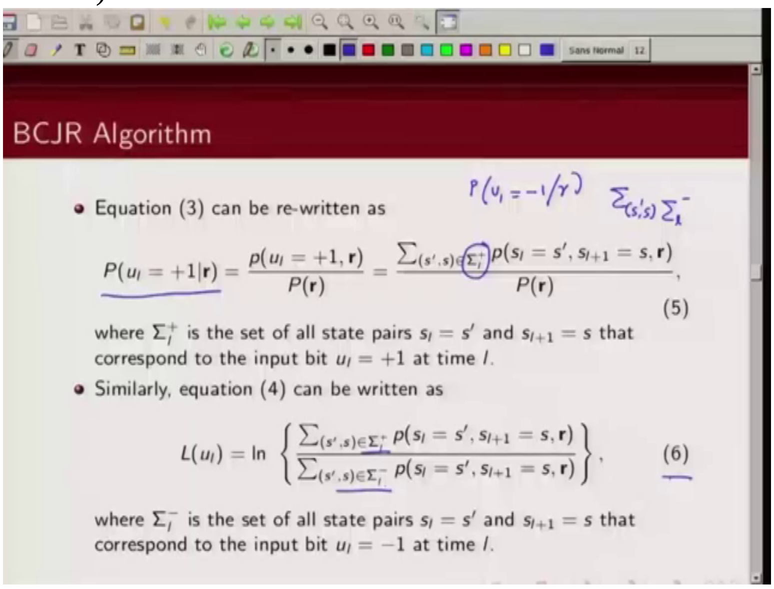


So this, the summation is now only over valid transitions corresponding to u l being plus 1 and this summation is over valid transitions corresponding to u l being minus 1.

(Refer Slide Time 12:41)

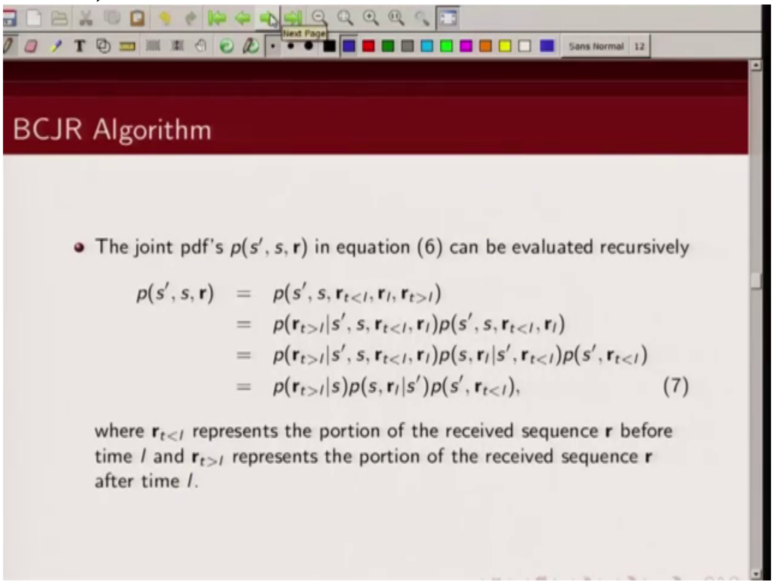
So I have simplified my equation number 4

(Refer Slide Time 12:45)



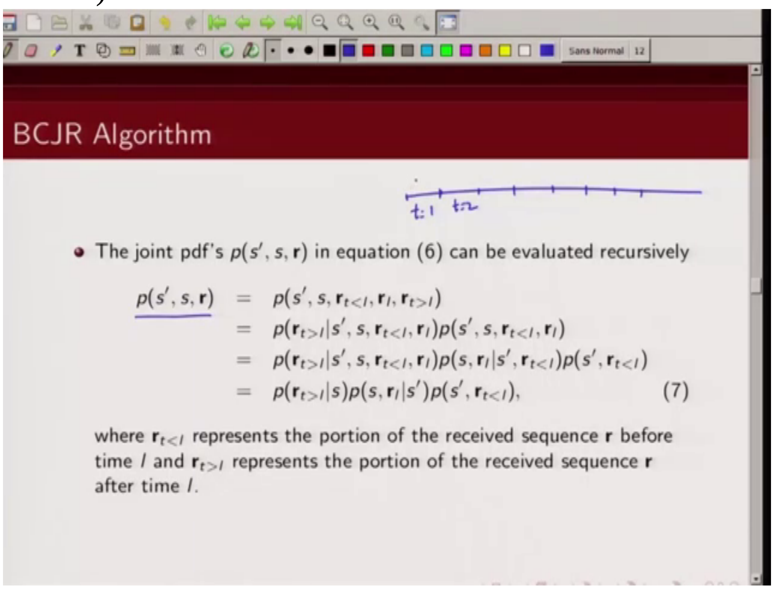
in equation number 6 and I have used the state diagram or the Trellis diagram of the convolutional encoder to simplify my expression. So this will be my a posteriori probability log likelihood L value a posteriori probability. Now how do I compute this term? This we will show that if we can write this term as product of three terms and two of these terms can be computed recursively that's what I am going to show in the subsequent slide. So let us look at this expression. How do we compute the probability that in the current state it is in s prime, the next state is s given a recieved sequence r? So as I said

(Refer Slide Time 13:35)



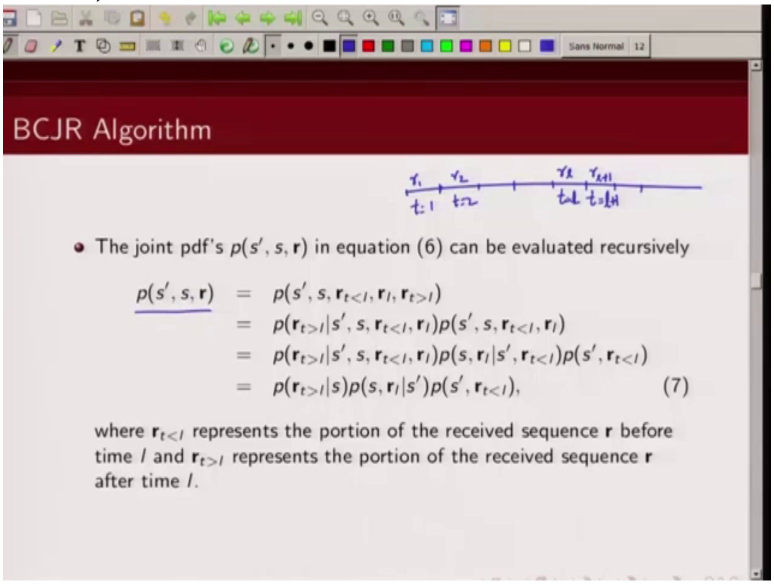
we are interested in this. Now this can be written as, so I have this received sequence r. So let us say this is r at time t equal to 1, t equal to 2 so this is my let us say time instances and I get some bits, let us say I get some

(Refer Slide Time 13:56)



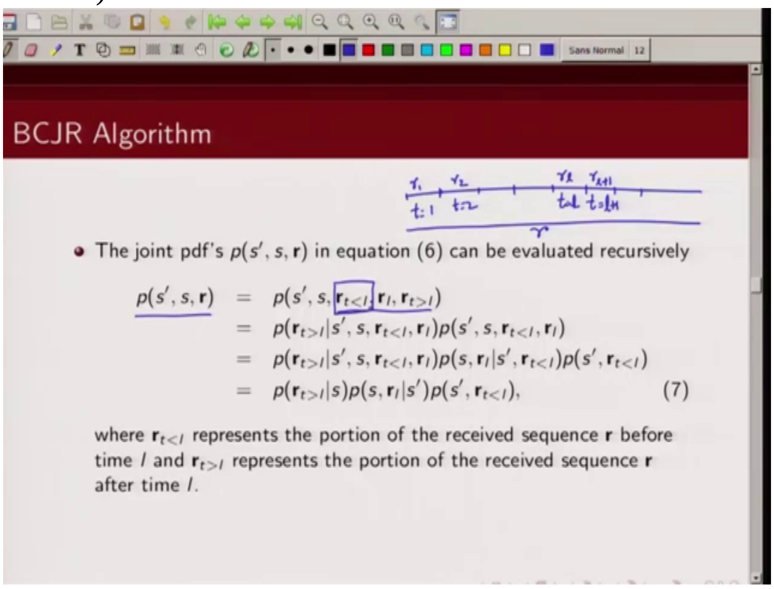
r 1 corresponds to what I receive at time t equal to 1, r 2 corresponds to what I receive at time 2, r l corresponds to what I receive in time l and like that, r l plus 1 is what I receive at time t equal to l plus 1, like that. So this

(Refer Slide Time 14:17)



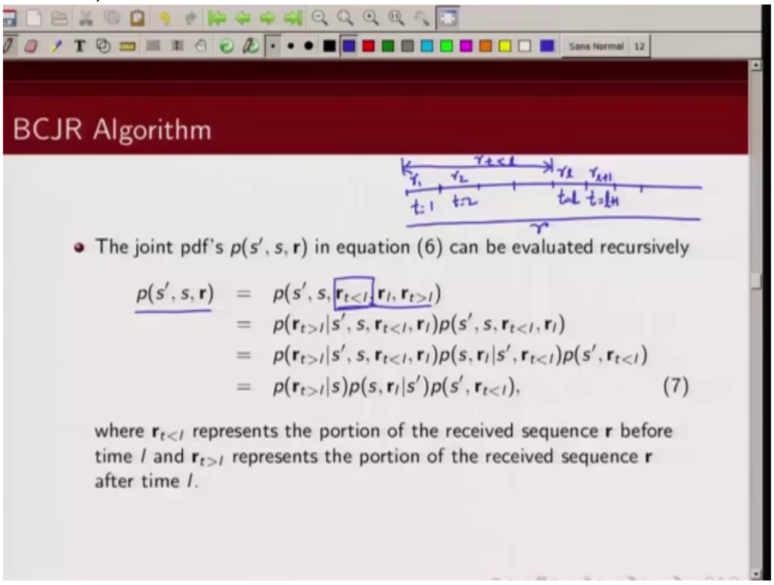
received is, whole thing is my received sequence r, Ok. Now what I am doing is I will partition that received sequence into 3 segments. So one, which corresponds to, one is this,

(Refer Slide Time 14:36)



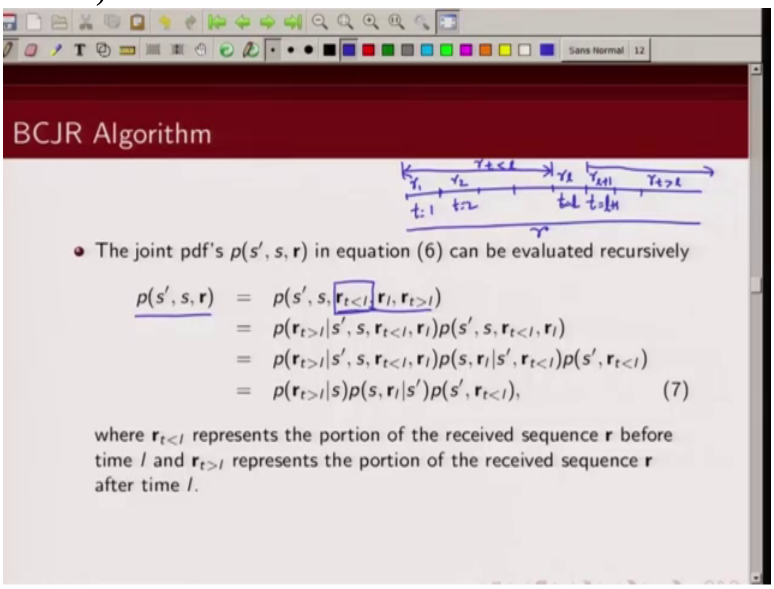
which corresponds to time before l. So one is this portion, this portion of my recieved sequence. This is r t less than l. Next

(Refer Slide Time 14:50)



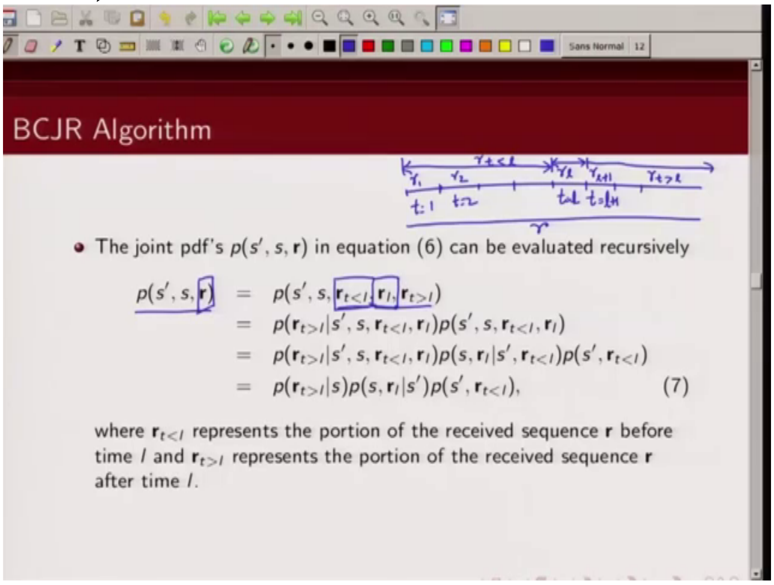
is this section which corresponds to r t greater than l and then

(Refer Slide Time 14:58)



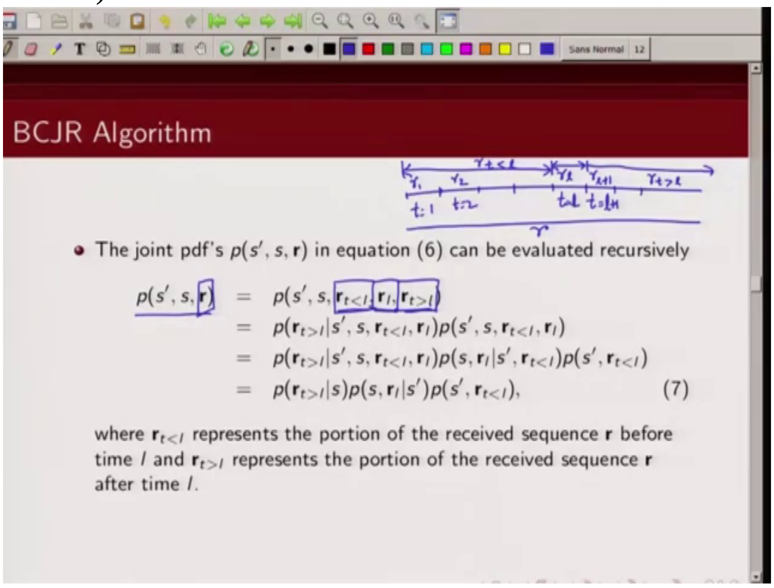
third section is this, which corresponds to r l, Ok. So what I did was I split this r into 3 segments. One is r corresponds to time less than l, r at time l and

(Refer Slide Time 15:18)



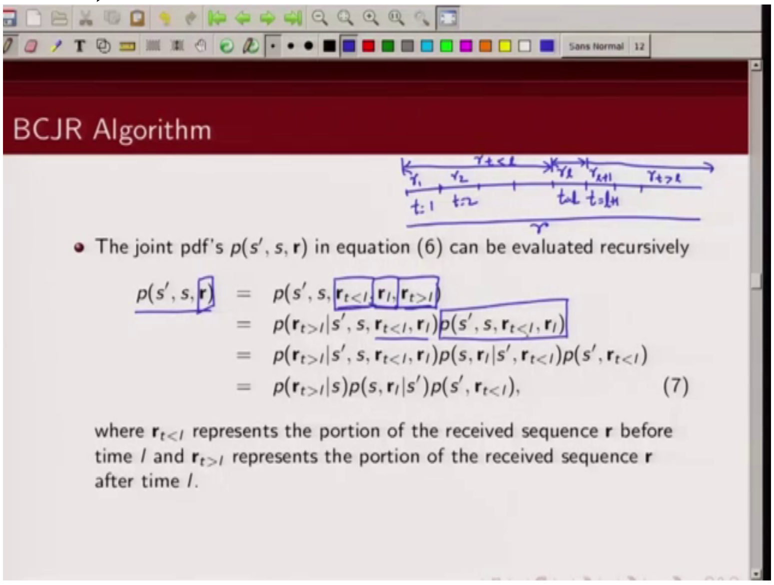
r at time greater than

(Refer Slide Time 15:21)



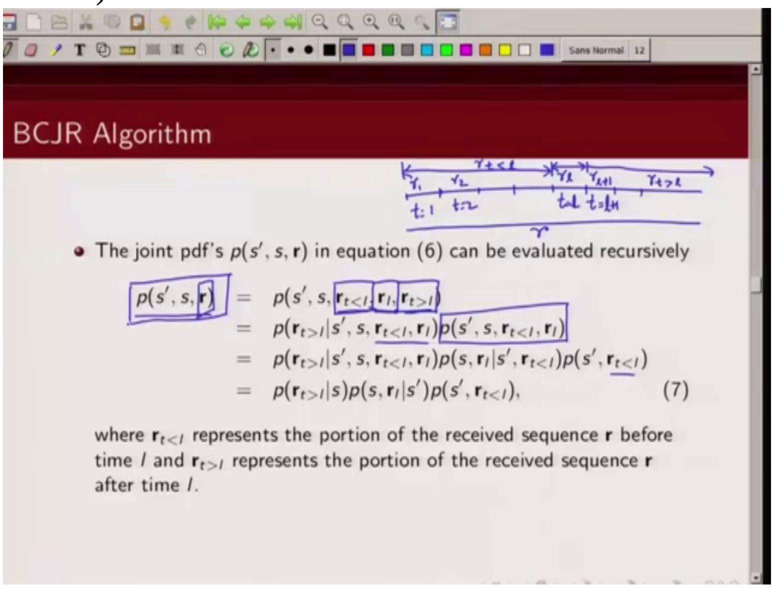
l. Now using base rule I can write this probability as probability of r at time greater than l given s prime and s and this r into probability of s prime s and r at t less than l and r l. Now subsequently I can further simplify this, again apply Bayes rule and I can write

(Refer Slide Time 15:54)



this as probability of s and r l given s prime and r t less than l into probability of s prime and r t less than l. So note now this term that I had here, so applying Bayes rule essentially,

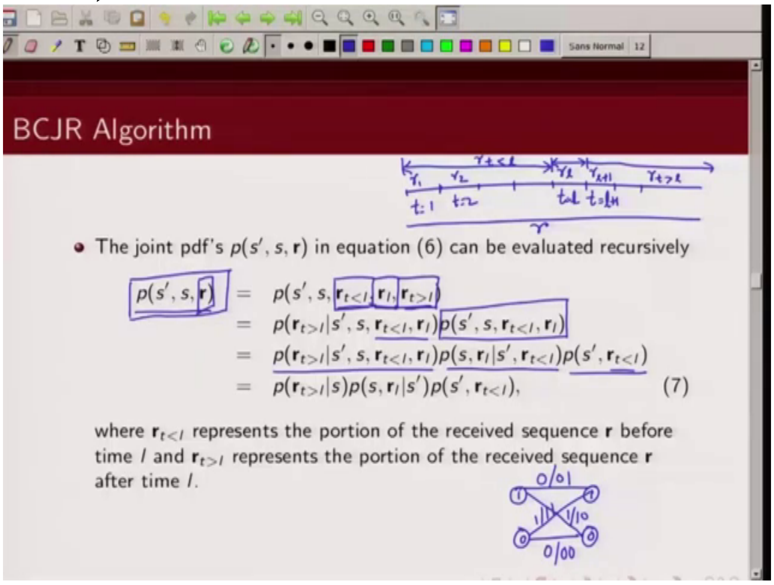
(Refer Slide Time 16:17)



I broke it up into 3 terms. One is this term, second is this term and third is this term, Ok. Now let's look at this. So probability of r when t is greater than l given initial state s prime, next state s and the recieved sequence before l and recieved sequence is l. So let us look at the Trellis diagram. Let's go back and look at the Trellis diagram at time l.

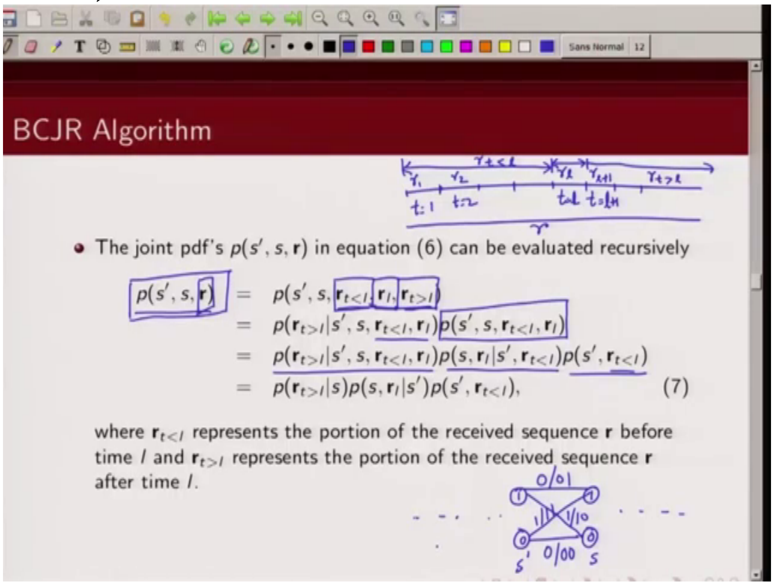
Let's take this example of 2 state code. So what I had was 0 by 0 0, 1 by 1 1 then I had this, 1 as 1 0 and this was 0 by 0 1.So this was my Trellis diagram. This is all zero state; this is state 1, Ok.

(Refer Slide Time 17:16)



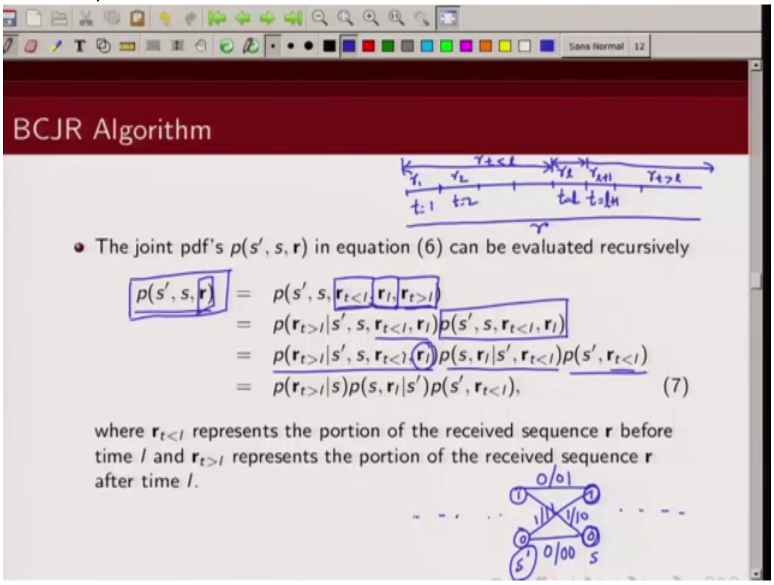
Now note and like that you have, you have, in Trellis diagram you have, this is one Trellis section. You will similarly have Trellis sections others. So this is a time l. So you are interested in what is the probability of r t greater than l given previous state s prime given next state s given the received sequence before time

(Refer Slide Time 17:45)



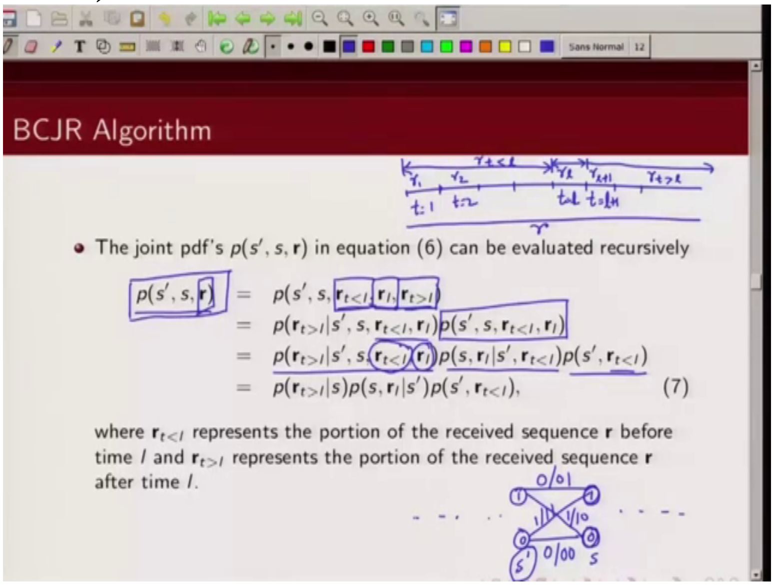
t equal to l and given the current received sequence. Now note that if I specify this next state, so probability of r t greater than l given s then I don't need information about the previous state. I don't need information about what is the current input, I don't

(Refer Slide Time 18:12)



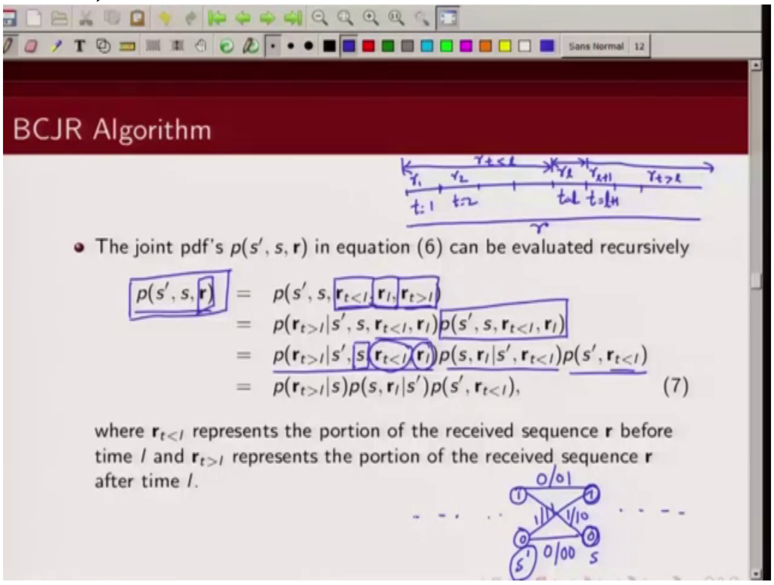
need information about what was the received

(Refer Slide Time 18:15)



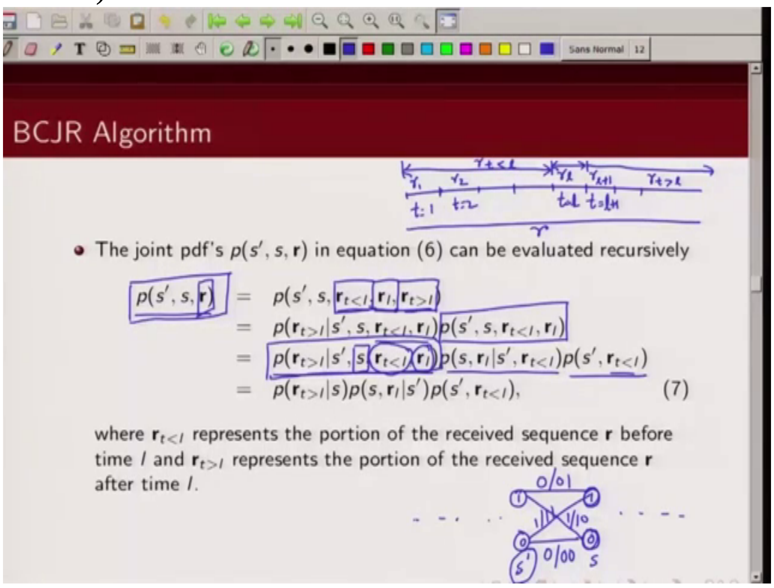
sequence before l provided I know what is the next state s.

(Refer Slide Time 18:21)



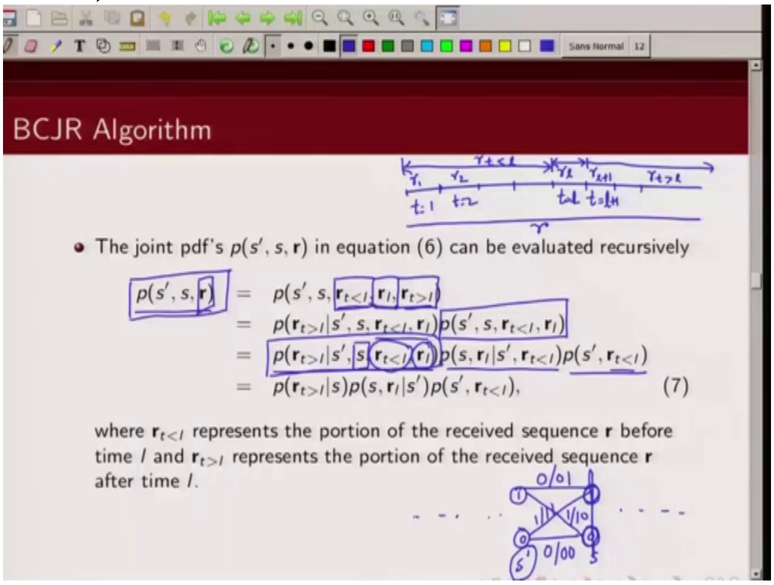
So this probability that you see here, probability of r t greater

(Refer Slide Time 18:28)



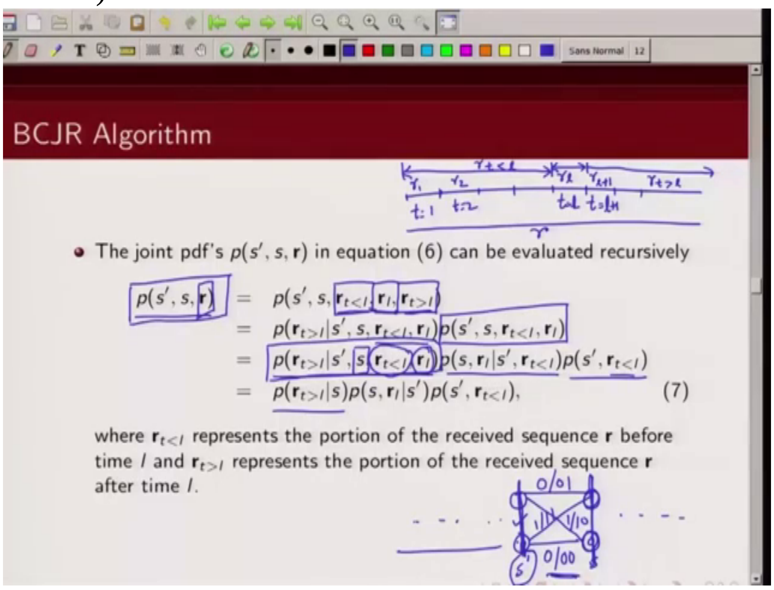
than l given s prime s and this received sequence r can be then written as probability of r t greater than l given only s because knowing this final state s I don't need information about what was my state

(Refer Slide Time 18:47)



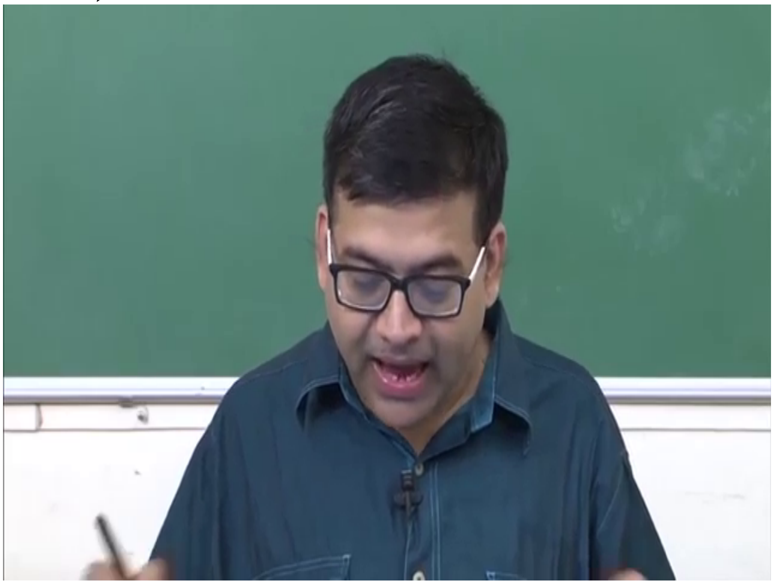
here. I don't need information about what my received sequence was here. I don't need information about what my past recieved sequence was, provided I know what was my next state s. So this, given these quantities will only depend on s. So I can simplify this expression like this. The same thing here, look at probability of being s r l given previous state and given the input before time t equal to l. Now if I specify what the previous state is, then I don't

(Refer Slide Time 19:25)



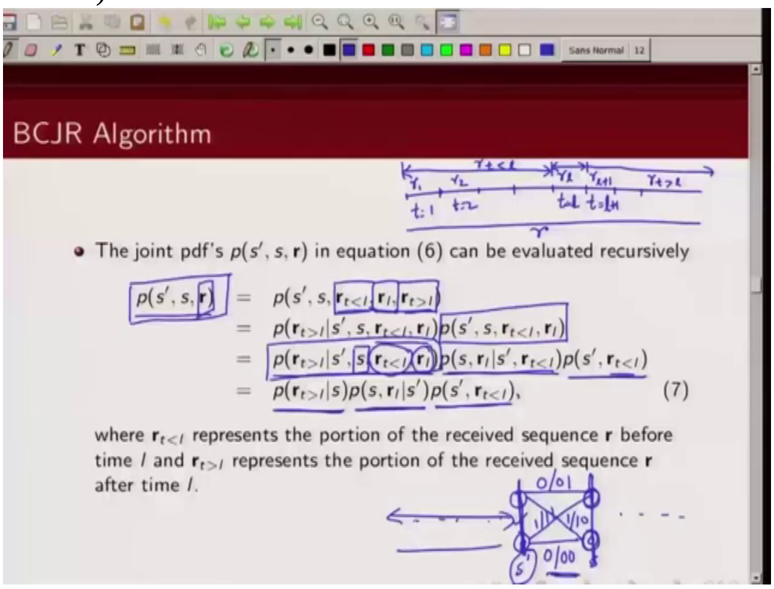
need what was my input at time t less than l. So this can be simplified into this expression. And then of course we have this third expression which is this. So what we have done is this joint probability we have now split up into three probabilities, one is this, second one is this, and third one is this, Ok

(Refer Slide Time 19:57)



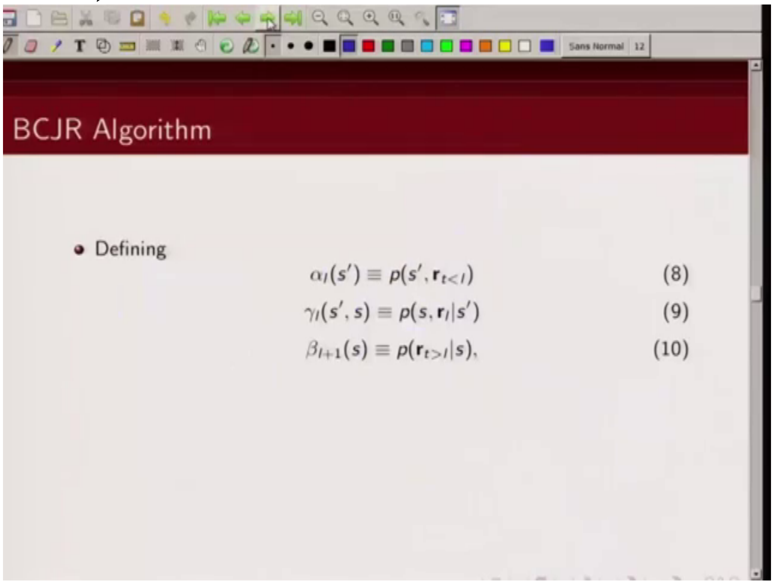
and we will now show how we can compute each of these

(Refer Slide Time 20:02)



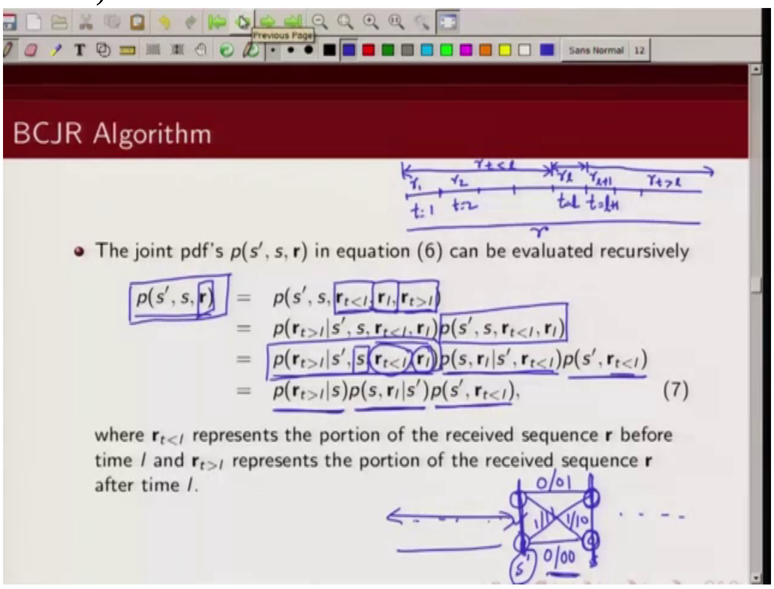
terms. So let us call

(Refer Slide Time 20:05)



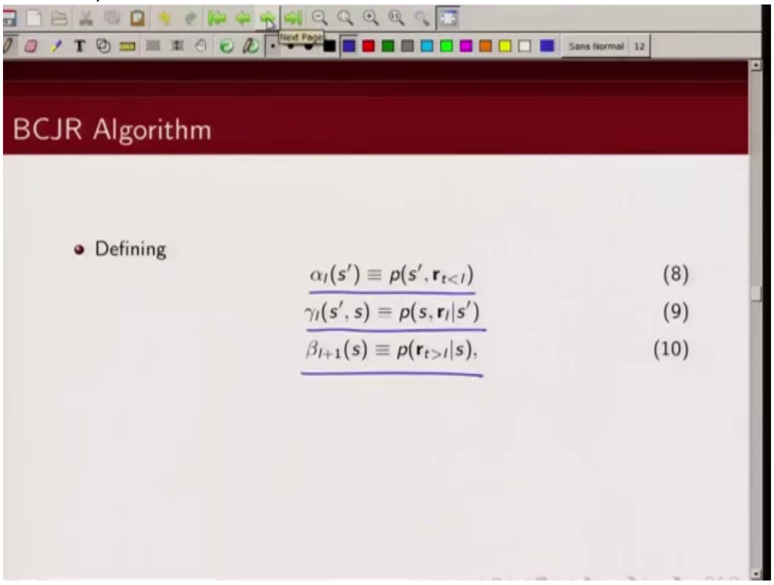
this probability by alpha, this probability by gamma and this probability by beta. And now we are going to show then we can write then this joint

(Refer Slide Time 20:20)



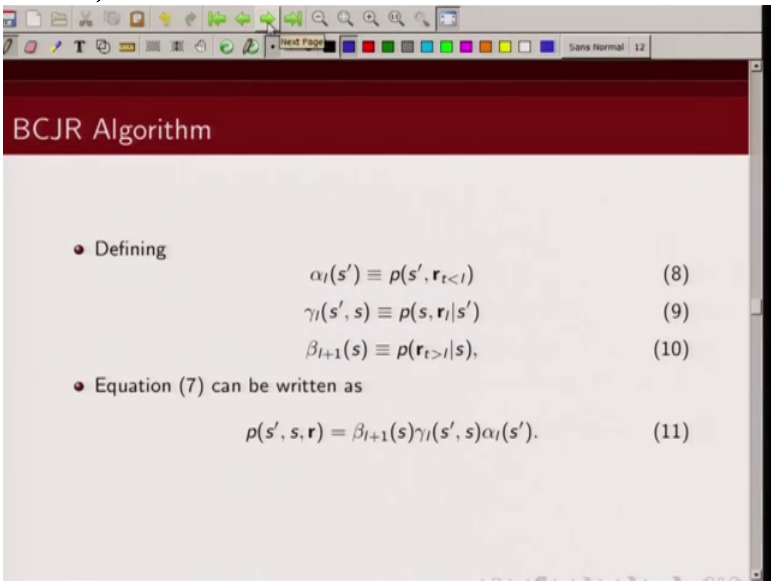
probability in terms alpha, beta and gamma.

(Refer Slide Time 20:25)



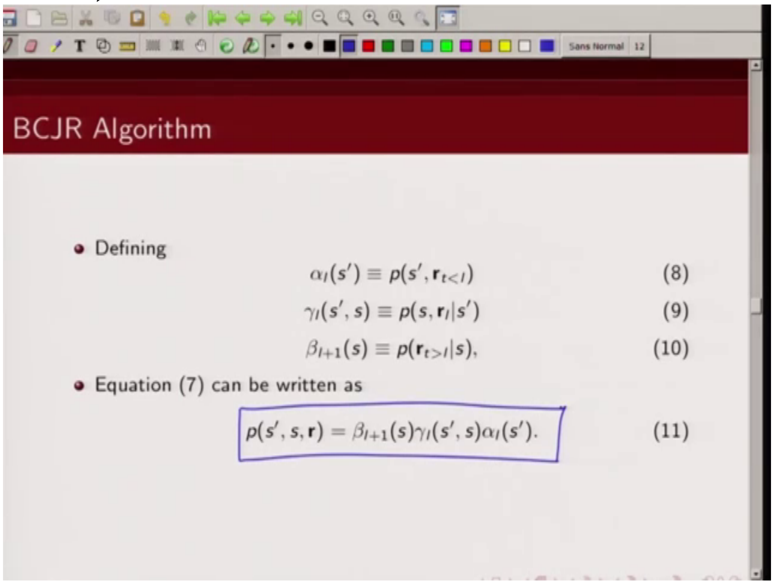
So

(Refer Slide Time 20:26)



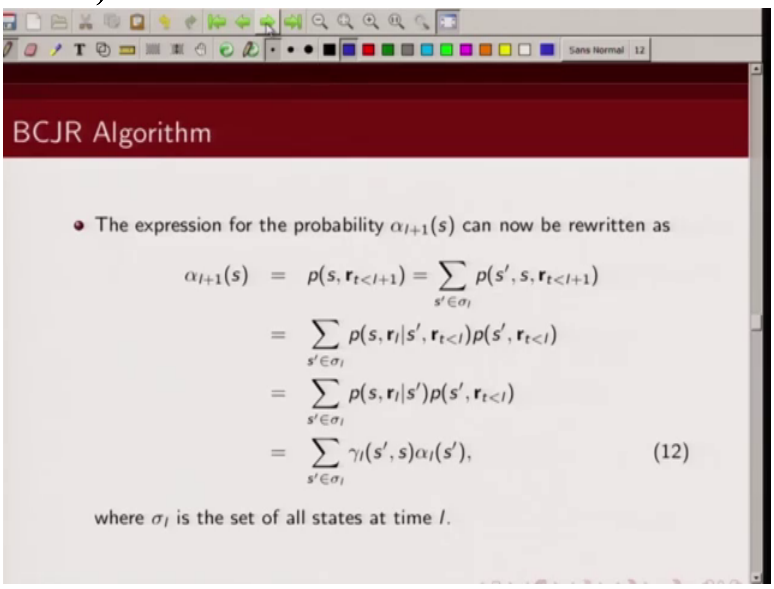
we can now write our equations in terms of alpha, beta and

(Refer Slide Time 20:32)



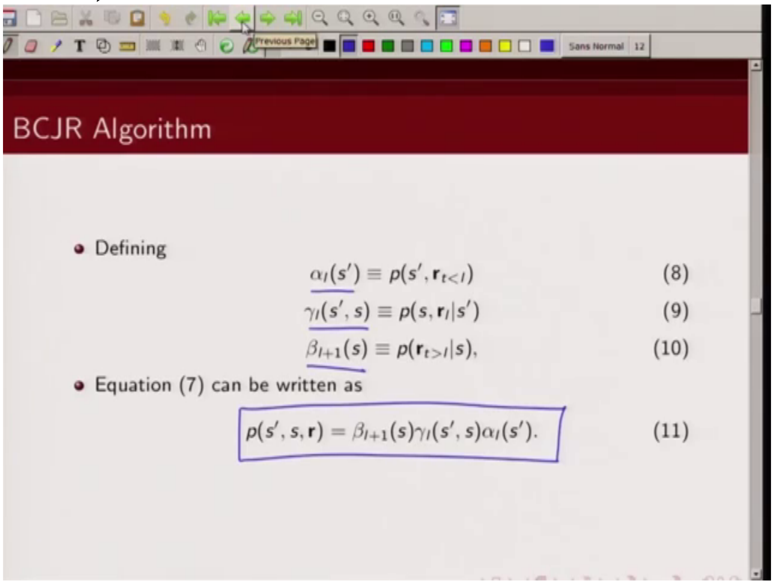
gamma, Ok. So let us now talk about how we can compute alpha, beta and gamma. So these

(Refer Slide Time 20:46)



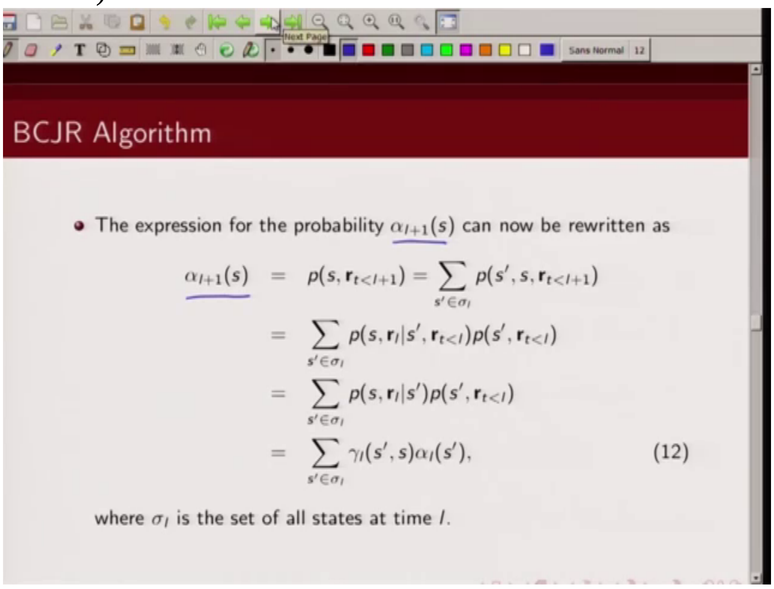
alphas can be computed using forward recursion as follows. So let us look at what is alpha plus 1 s. Now go back to our definition. So probability,

(Refer Slide Time 20:59)



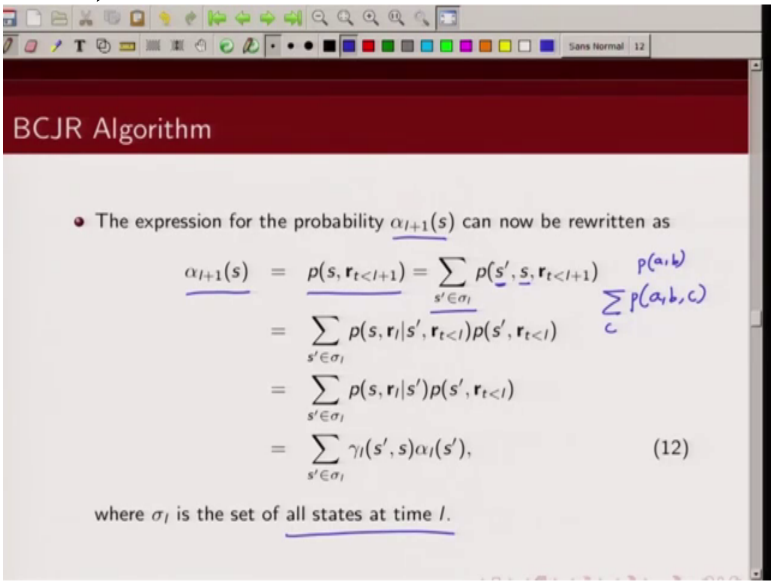
joint probability of being in state s prime and received sequence at time t less than l.

(Refer Slide Time 21:09)



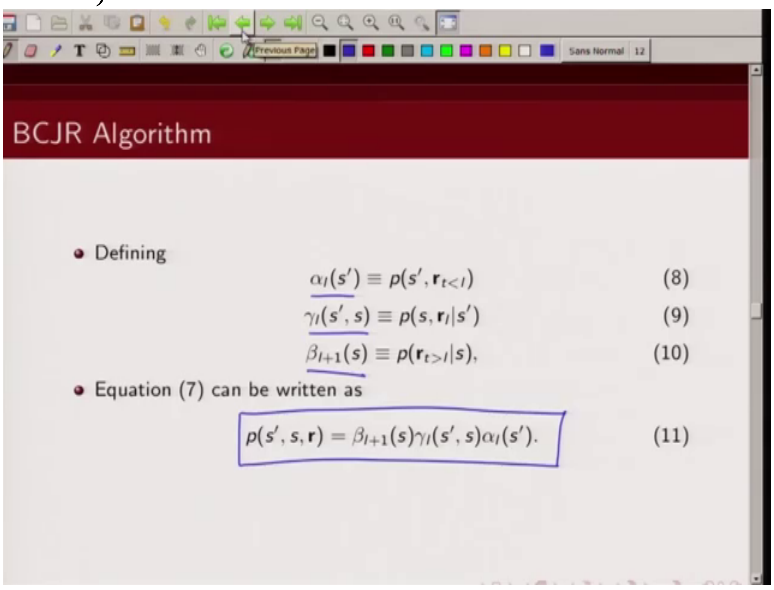
So alpha l plus 1 from definition, can be written like this. Now I can write this as, so I am adding a new variable which is the next state s and I am adding a new variable which is next state, previous state s prime and summing over all previous state. So what is this, summation over s prime belonging to all possible state at time l? So what I did was I had some term, probability term, probability of let's say a b, and what I did was I just added a term probability a b c and I summed over all possible values of c. So that's what I did here. I introduced a new variable

(Refer Slide Time 21:55)



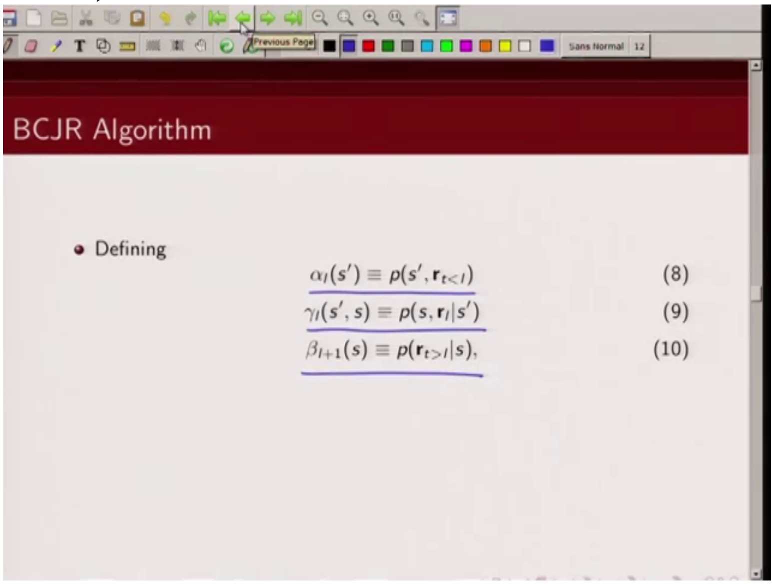
s prime and I summed over all these probabilities, all these possible values of s prime. Now this term can be written as product of these two terms, this is following exactly the same procedure which

(Refer Slide Time 22:16)



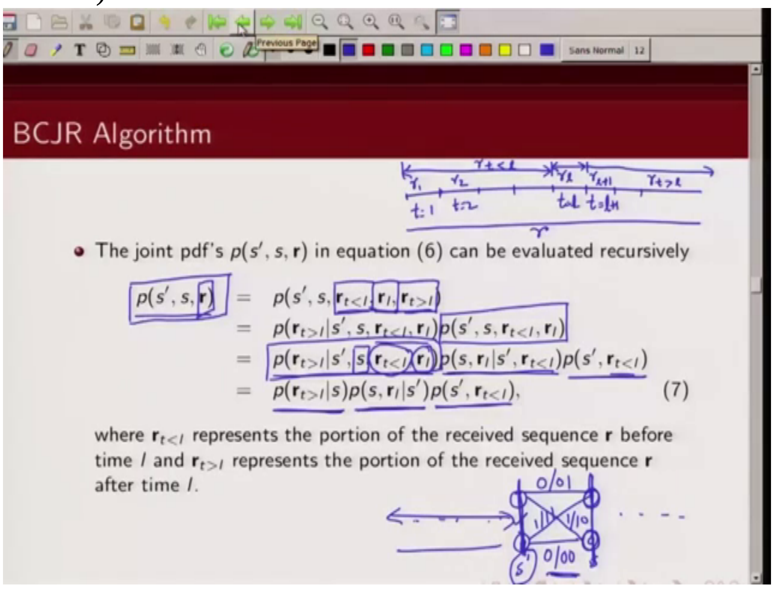
we followed

(Refer Slide Time 22:17)



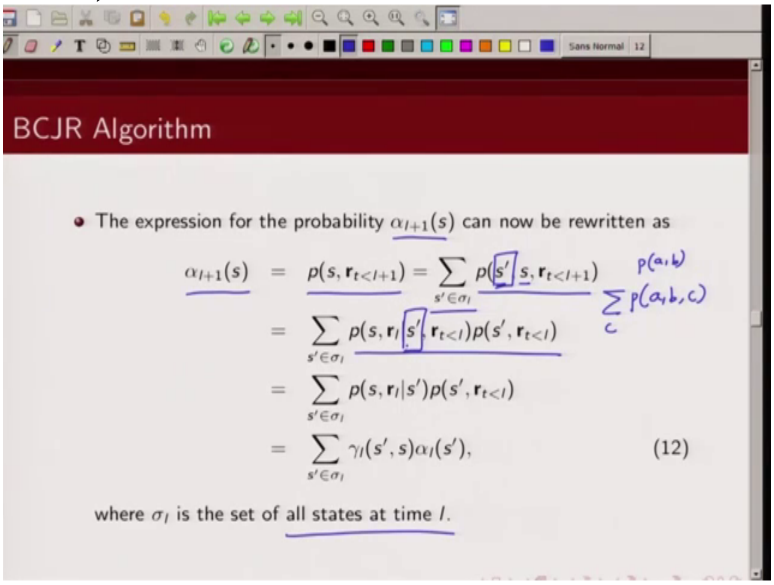
here.

(Refer Slide Time 22:19)



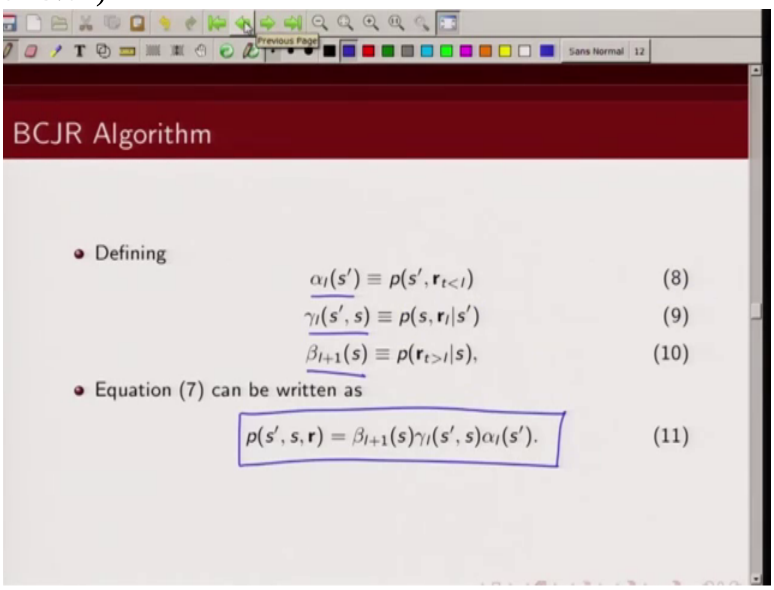
When we wrote this, we are basically using Bayes rule, now using Bayes rule, I can write this probability as product of these two probabilities. Now again the probability of s and r l given s prime and recieved sequence at time t less than l, if you know the previous state s prime you don't need

(Refer Slide Time 22:49)



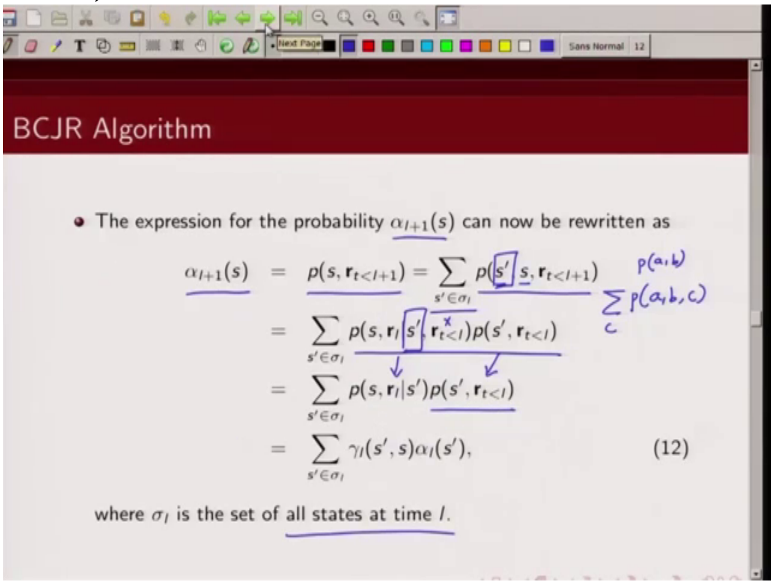
this information. So then this probability can be simplified to this probability and this is this. Now what is this term? This term is basically by definition

(Refer Slide Time 23:04)



our alpha and what is the next term, this is our gamma. So what I have shown you here

(Refer Slide Time 23:11)

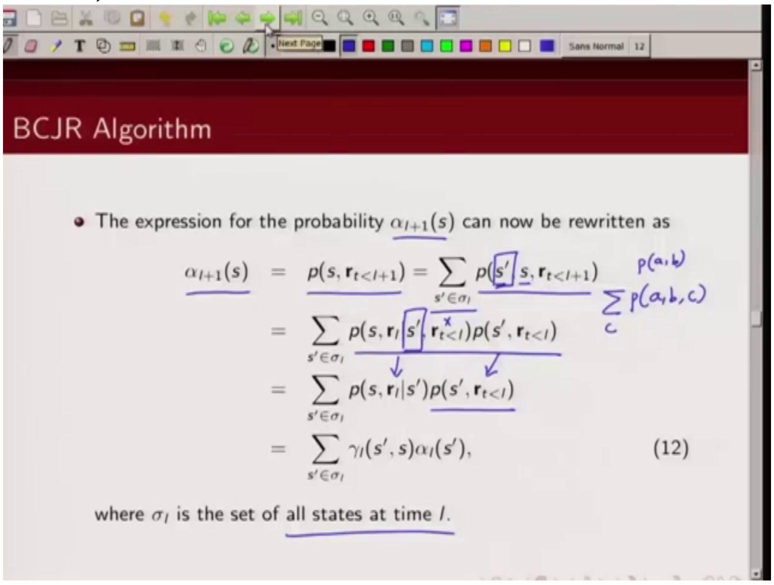


then is alpha at next state s

(Refer Slide Time 23:16)

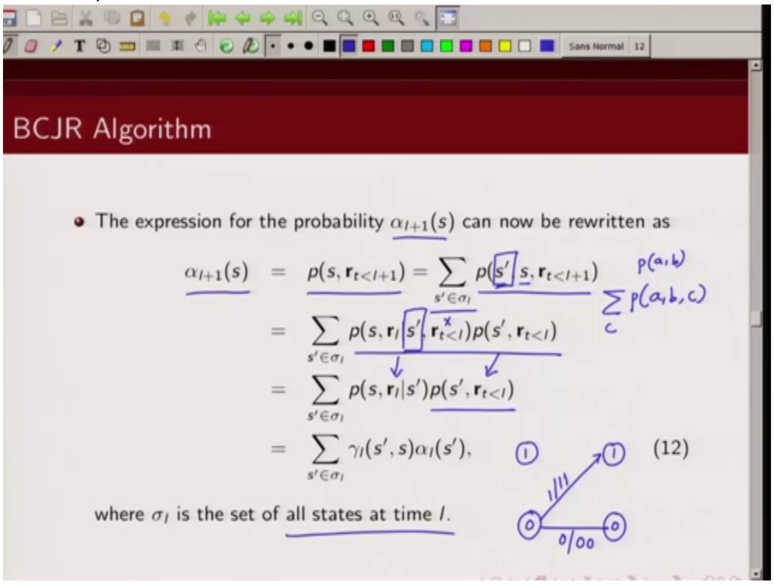
can be written as, can be computed recursively from alphas at previous state in this particular fashion. So again

(Refer Slide Time 23:27)



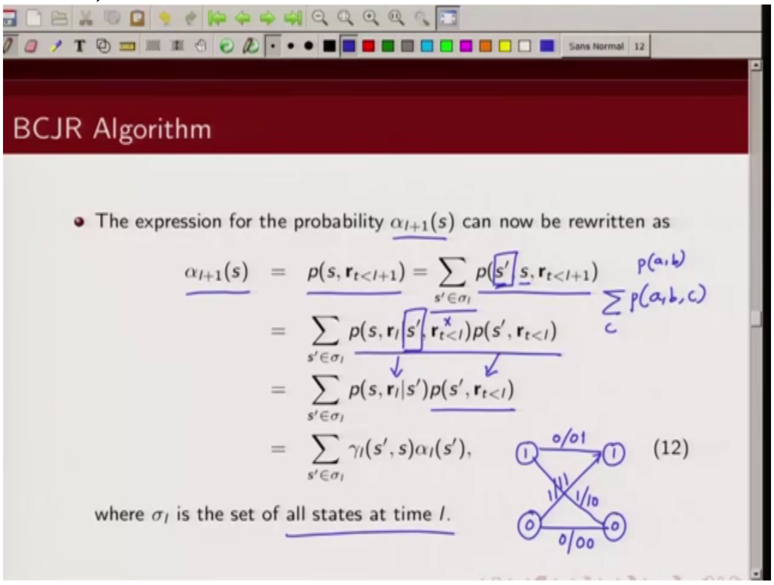
let's illustrate this with an example. Let's go back to our 2 state code example. So this is 2 state code. This is my all zero state. This is state 1, there are 2 transitions. Let's say this is 0 input, output 0 0, input 1, output 1 1,

(Refer Slide Time 23:49)



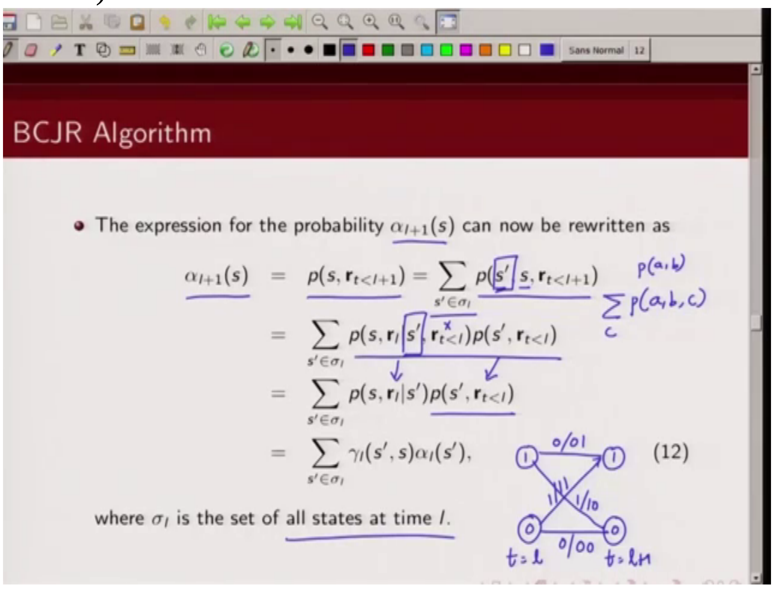
here input 1, output 1 0 and here input 0 and output 0 1.

(Refer Slide Time 23:57)



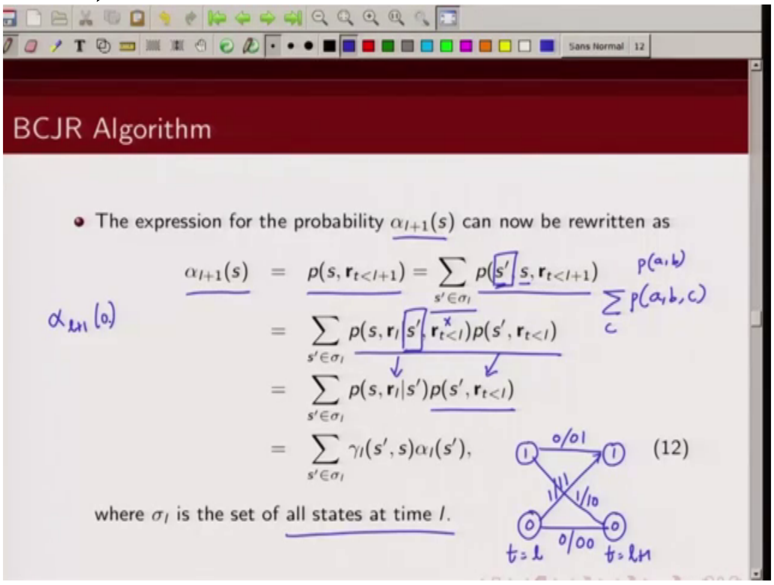
So then what would be the value of alpha? So let's say this is time t equal to some l and this is time t equal to l plus 1. So

(Refer Slide Time 24:10)



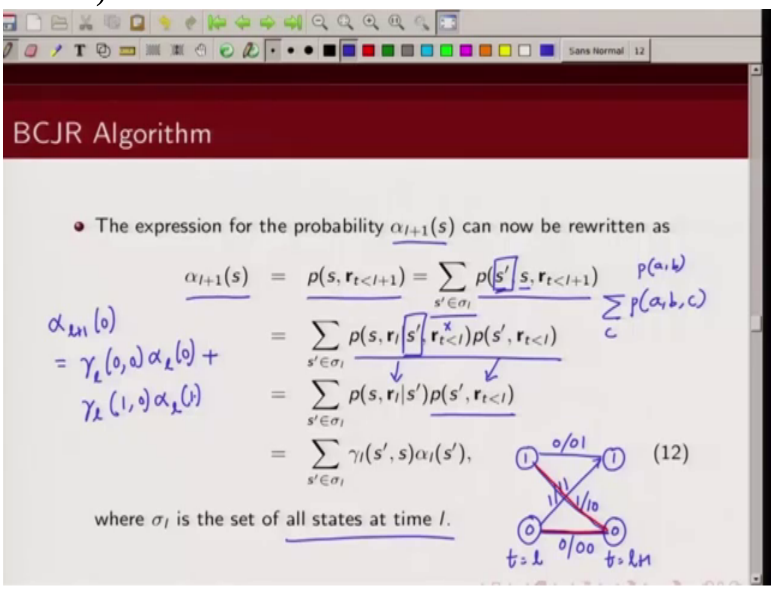
how can we write let's say alpha at l plus 1 for the state 0?

(Refer Slide Time 24:17)



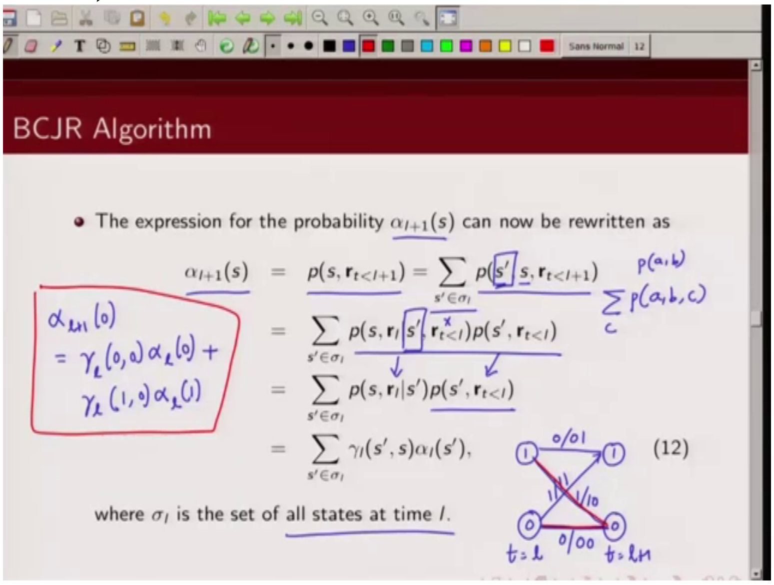
Now note here this is given by product of this summation over all input state right, now so alpha l 0 can be written as then gamma, this can be written as gamma at time l of 0 0. gamma l 0 0 is previous state is 0, next state is 0, gamma 0 0 into alpha l belonging to state 0. So this is gamma l 0 0, alpha l 0 so this is corresponding to this transition, Ok. This is corresponding to this transition, this term will come, fine. Now there is another transition here which is basically this. So we can write this will be plus gamma l 1 0 so gamma l 1 0 is the gamma corresponding to this transition when the initial state is 1 and next state is 0 multiplied by alpha at time l belonging to state 1, Ok . So alpha l plus l 0 can be then written

(Refer Slide Time 25:53)



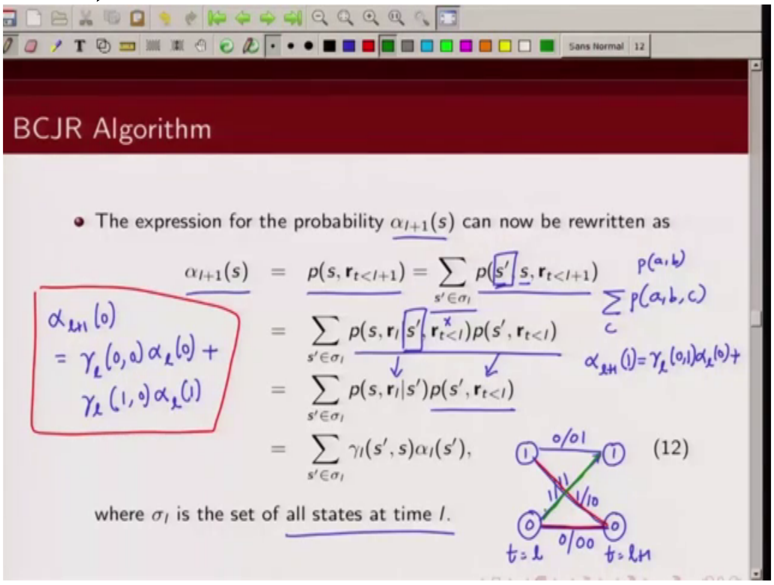
as this. Now similarly we can also compute what is the value of

(Refer Slide Time 26:01)



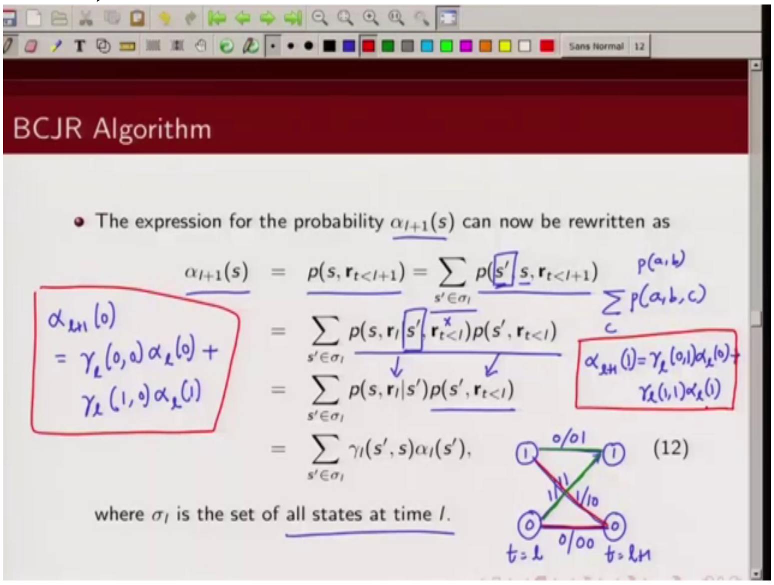
alpha l plus 1 at 1. So we repeat the same procedure. So let's write it here. alpha l plus 1 in the final state is 1 can be written as gamma l 0 1 times alpha l 0 plus, so this is corresponding to this transition, gamma l initial state 0, final

(Refer Slide Time 26:35)



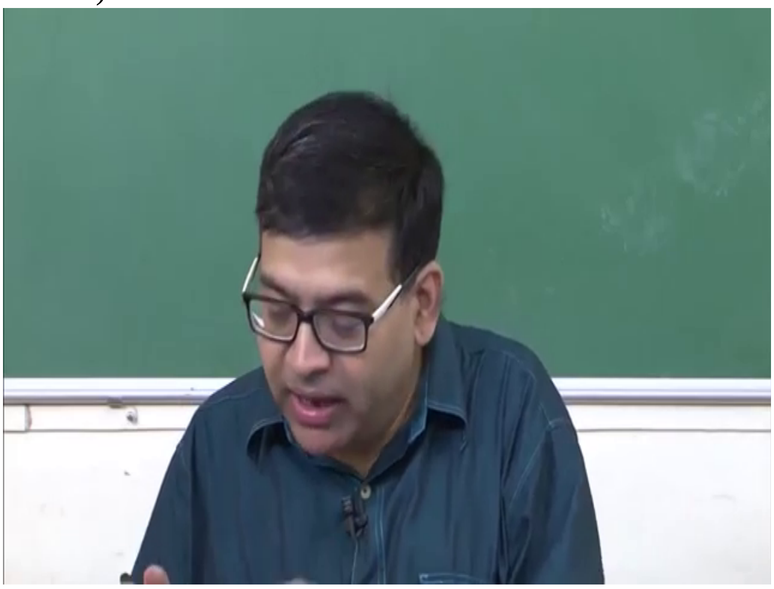
state 1 and alpha at time l 0 plus this another transition which is this. So this can be written as gamma l 1 1 times alpha l 1. So these are, for particular convolutional encoder

(Refer Slide Time 27:03)



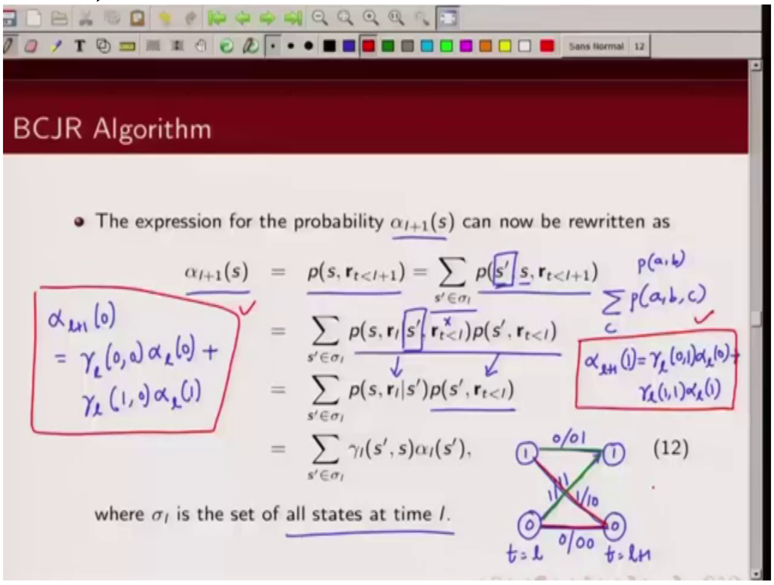
whose Trellis section is given by this, these are, these 2 are my alpha values, this one and this. So you see I can recursively compute alpha time

(Refer Slide Time 27:17)



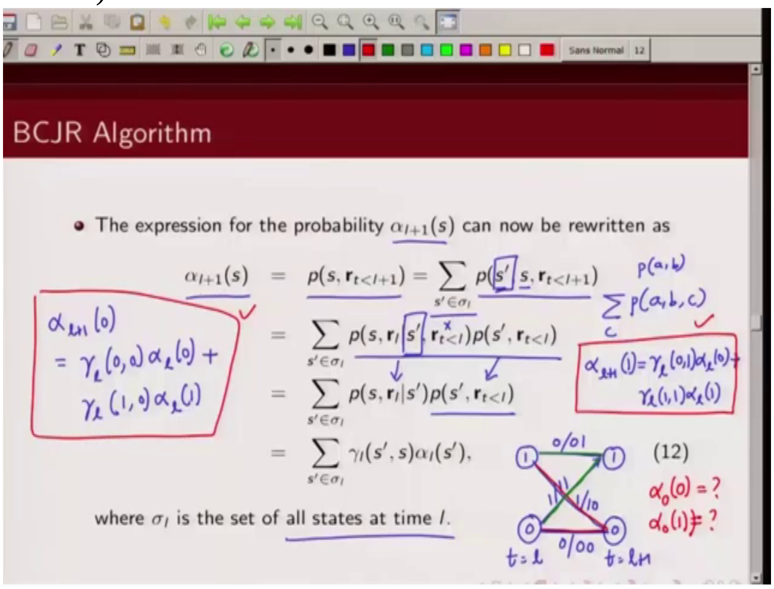
l plus 1 from alpha at time l and branch metric gamma. Now to do this recursion, we need to know what is the initial condition. What is the initial condition? We need to know

(Refer Slide Time 27:35)



what is the value of alpha 0 for different states, for state 0, for state 1. We need to know what the values of these are.

(Refer Slide Time 27:48)

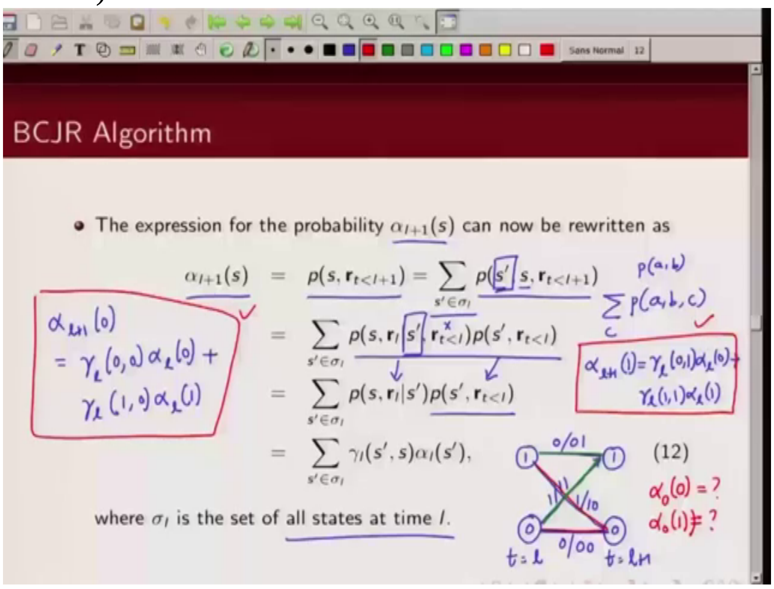


Now note initially we assume that the

(Refer Slide Time 27:52)

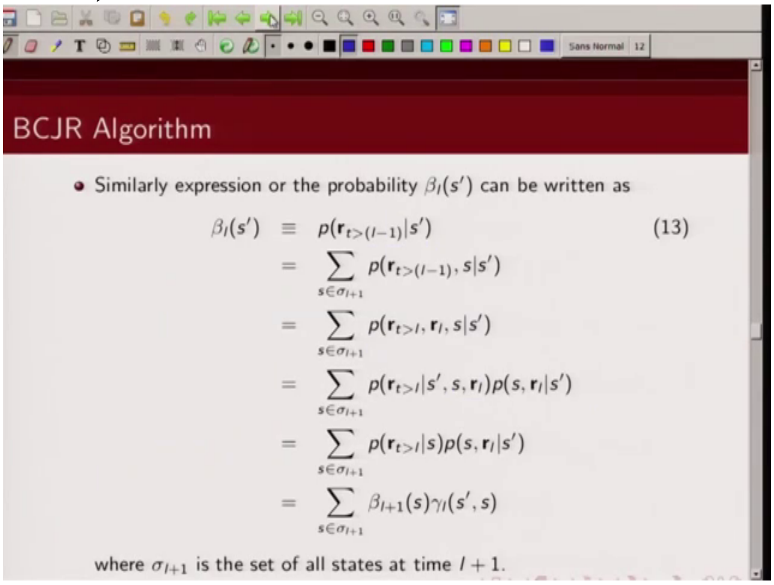
encoder is in all zero state. So if we assume the encoder is in all zero state then it is,

(Refer Slide Time 28:00)



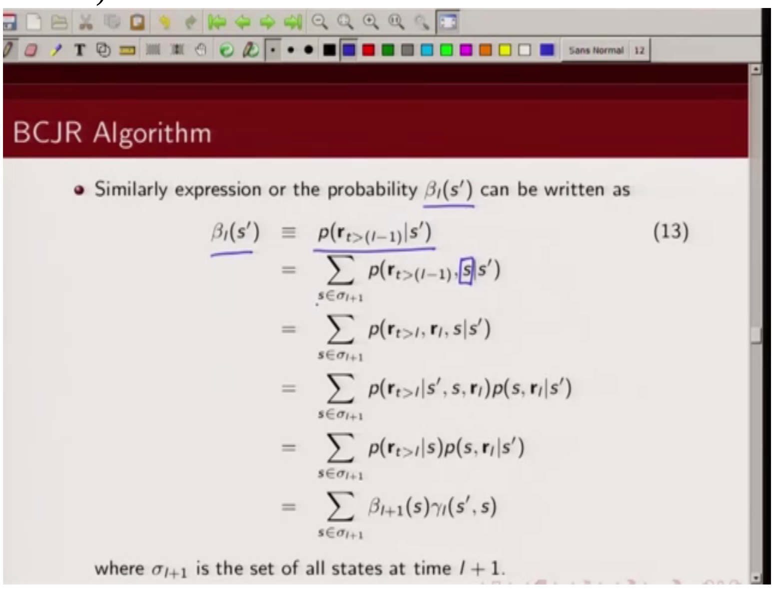
it is going to stay in all zero state then in that case, we consider this probability as 1 and this all other possibility of it staying all other state is 0. So the initial value when we assume that the encoder is in all zero state we assume that alpha 0 at 0 is 1 and alpha 0 at at any other state is 0. So similarly we can

(Refer Slide Time 28:26)



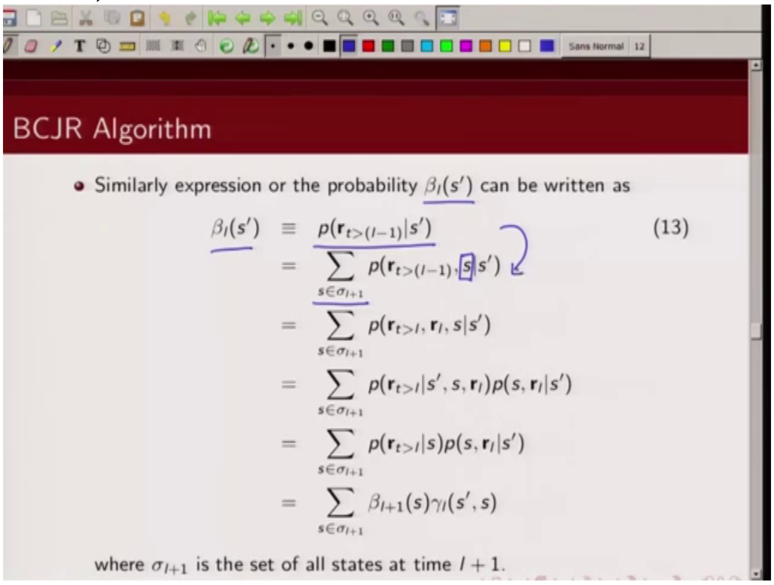
again we introduce a new variable s and sum over

(Refer Slide Time 28:46)



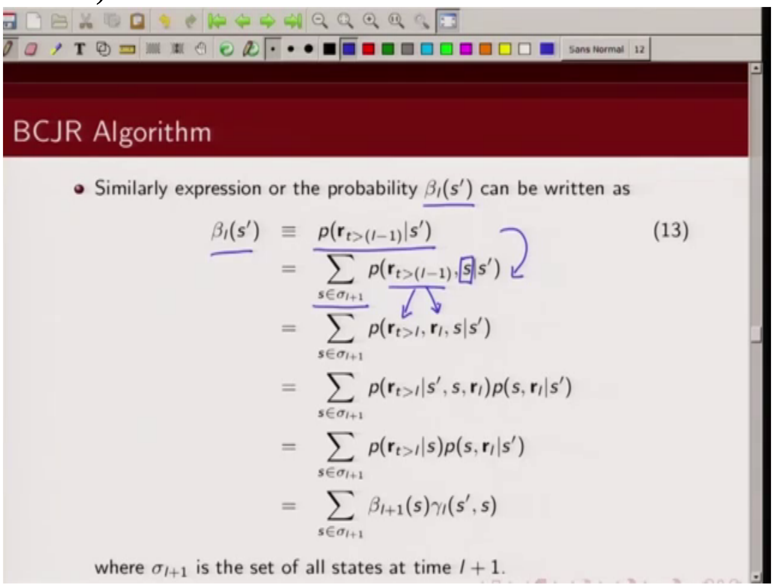
all possible values of s, so then this becomes, from here we get this. Now

(Refer Slide Time 28:55)



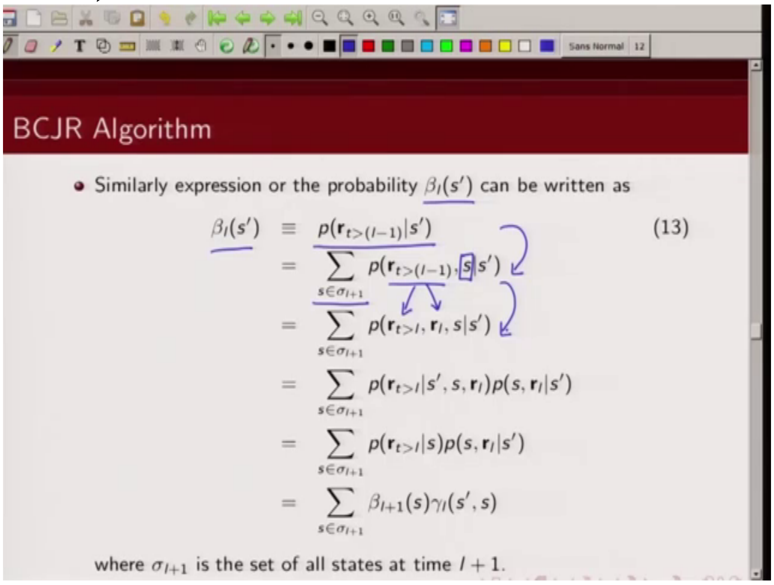
we split this r into these 2 terms,

(Refer Slide Time 29:01)



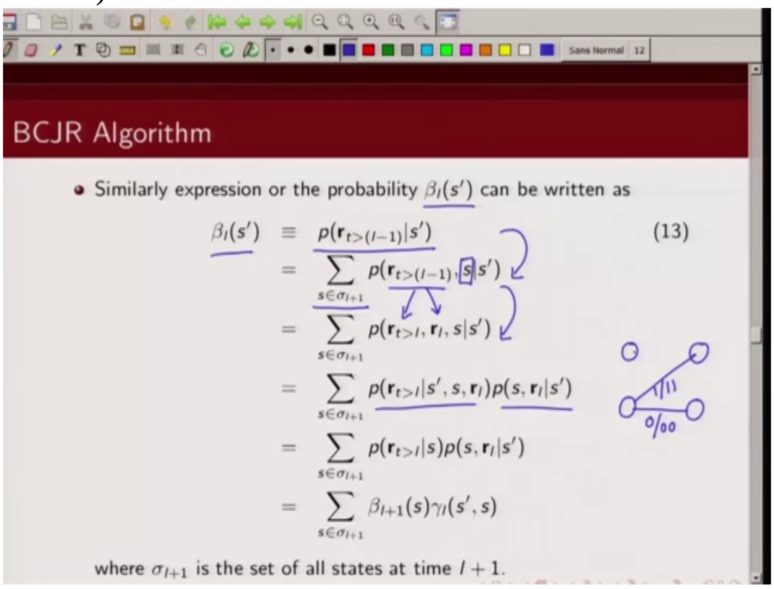
so we get this expression.

(Refer Slide Time 29:05)



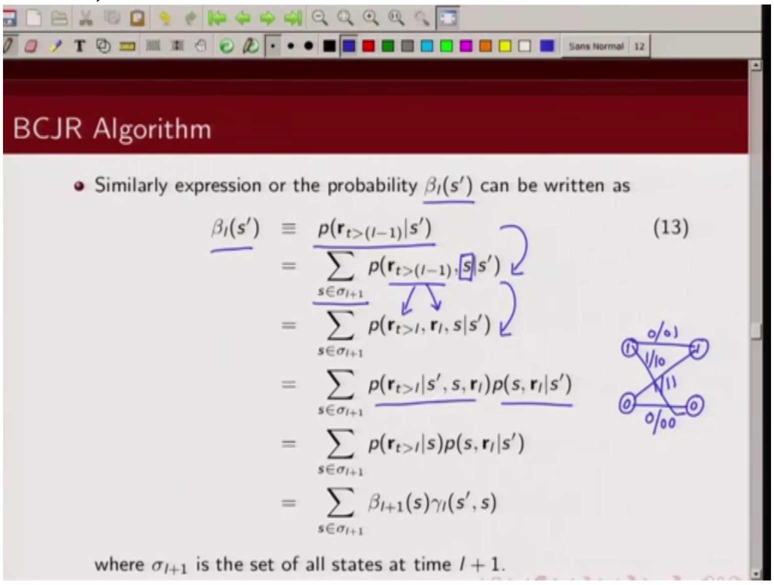
Now using Bayes rule I can separate out this term into 2 terms like this. And we know that, again let's go back to our Trellis section, so 0 0 1 1 1, this is

(Refer Slide Time 29:27)



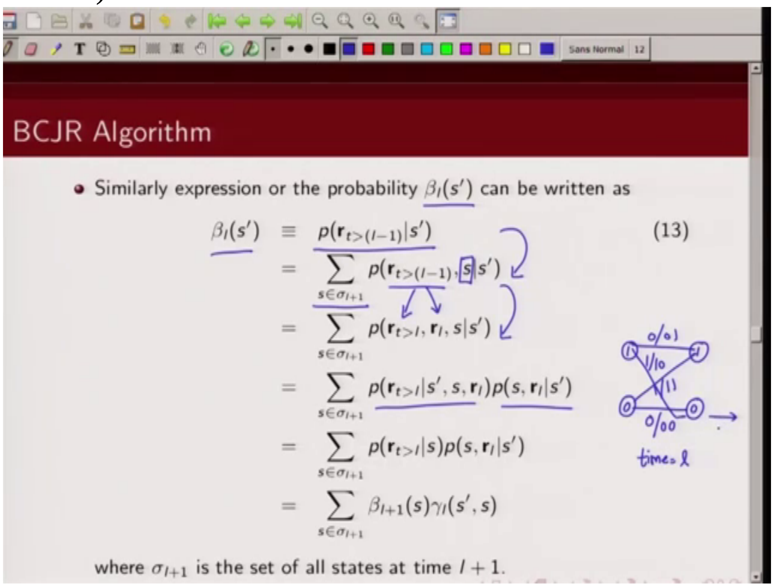
1 1 0 and this is 0 0 1. This is state 0, this is state 1. So if you are interested

(Refer Slide Time 29:37)



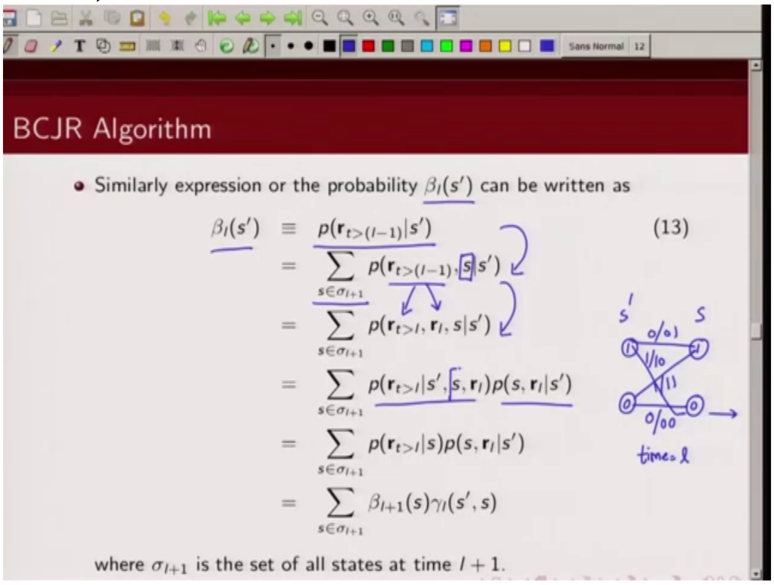
in probability of r t greater than l, that is this is your, this is your time l so probability of r t

(Refer Slide Time 29:50)



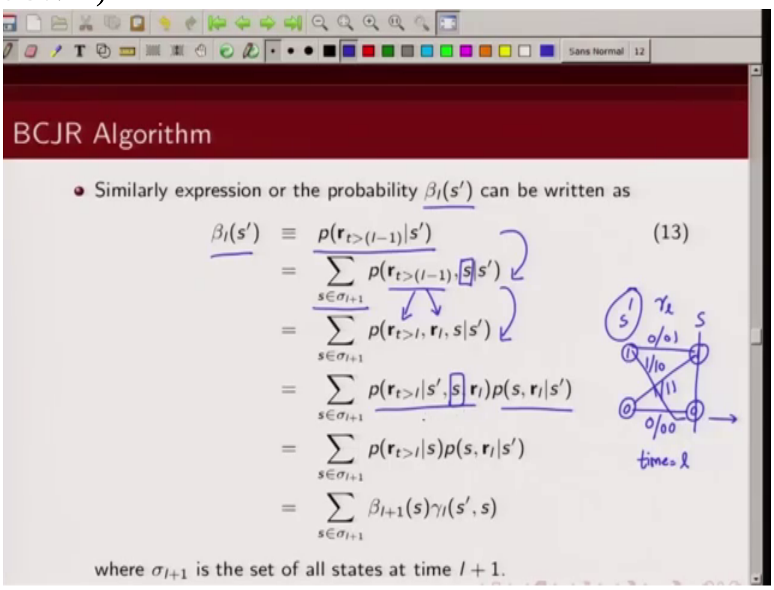
greater than l given previous state s hat, next state s and r l, it only depends on,

(Refer Slide Time 29:58)



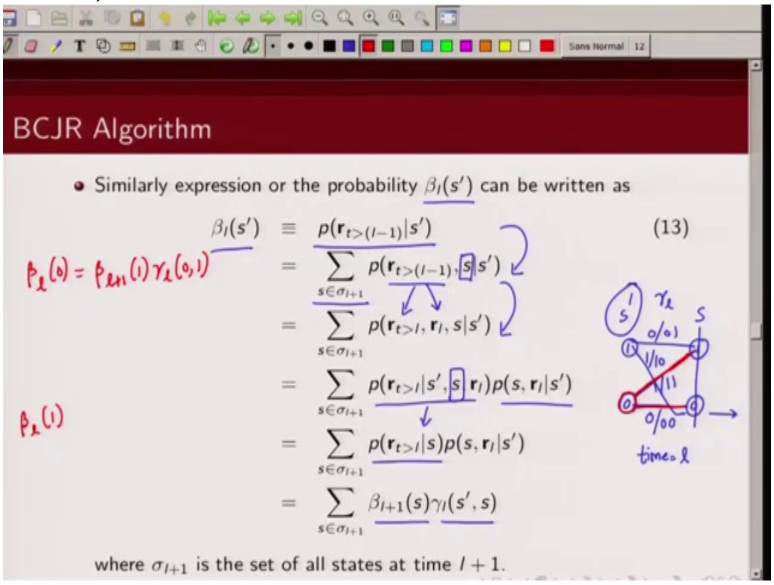
so if you know the next state s you don't need information about the previous state, you don't need information about the current bit. So I can simplify this expression

(Refer Slide Time 30:12)



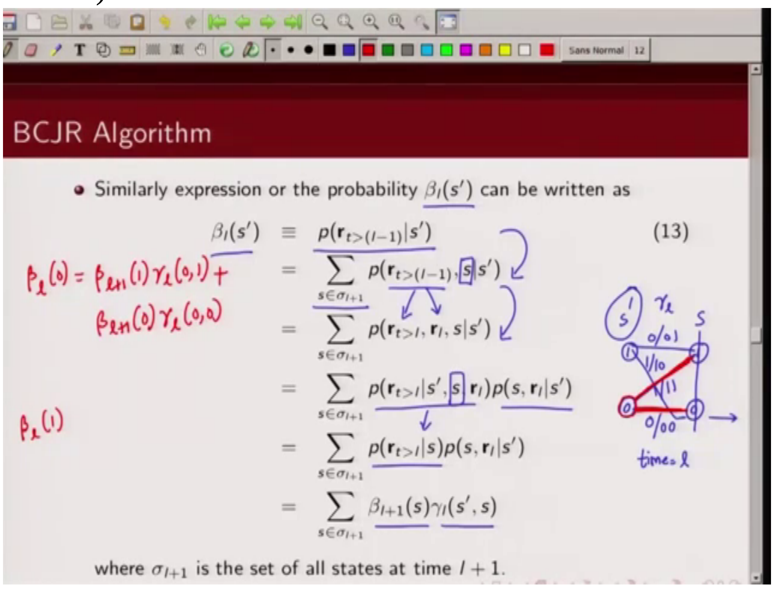
in this particular fashion and if we go back, this is nothing but our betas and this is our gamma. So let's compute beta for this particular code. So we are interested in computing beta l for 0 and beta l at state 1. So beta l at state 0 would be, so beta l at state 0, so we, so we are interested in computing beta at state 0, so this is sum over all those transitions which are ending at this state. So there are 2 transitions, one is this one, another is this one. So let's write the expression for this particular term. This we can write as beta at time l plus 1, 1 times gamma l 0 1. So the contribution of this is

(Refer Slide Time 31:21)



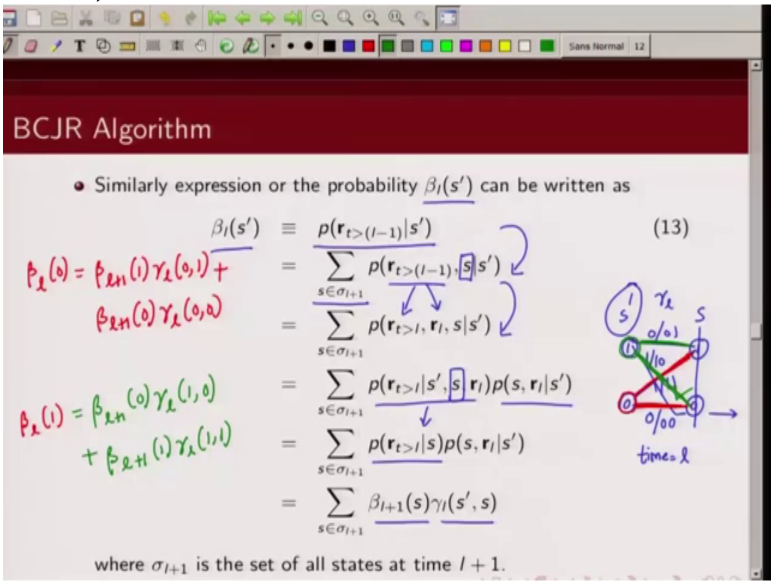
beta l plus 1 corresponding to state 1 multiplied by gamma of this Trellis section, gamma l when the initial state is 0 and the next state is 1. So this, this, this will contribute this term plus there is another transition which is this, this one right. So we can write contribution of this as beta l plus 1 zero times gamma l 0 0.

(Refer Slide Time 31:59)



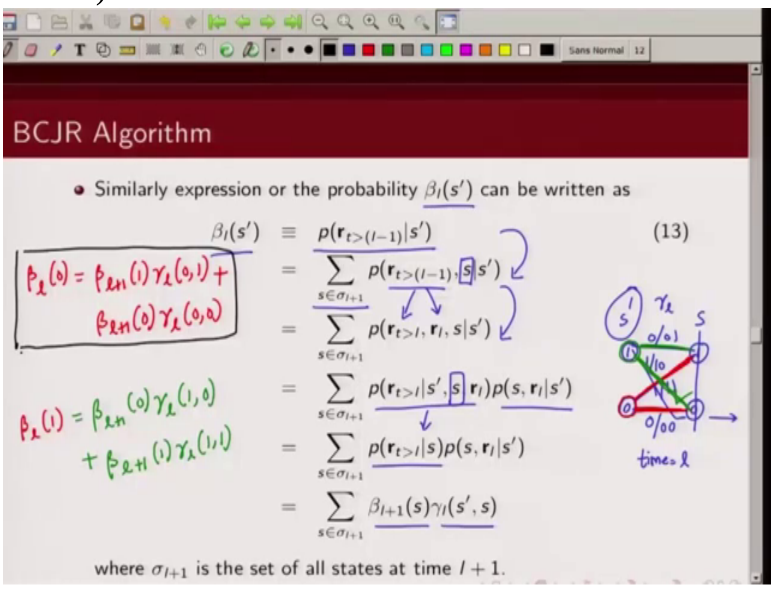
So this is our expression of beta l for state 0. Similarly we can compute beta l for state 1. So what are the 2 transitions which are ending at this state? One is this one, other one is this one. So let's write down the expression for this one, this one. So this will be beta l plus 1, 0 times gamma l 1 0 this is this term and what about this particular term, this will be given by beta l plus 1 1 times gamma l 1 1.

(Refer Slide Time 32:55)



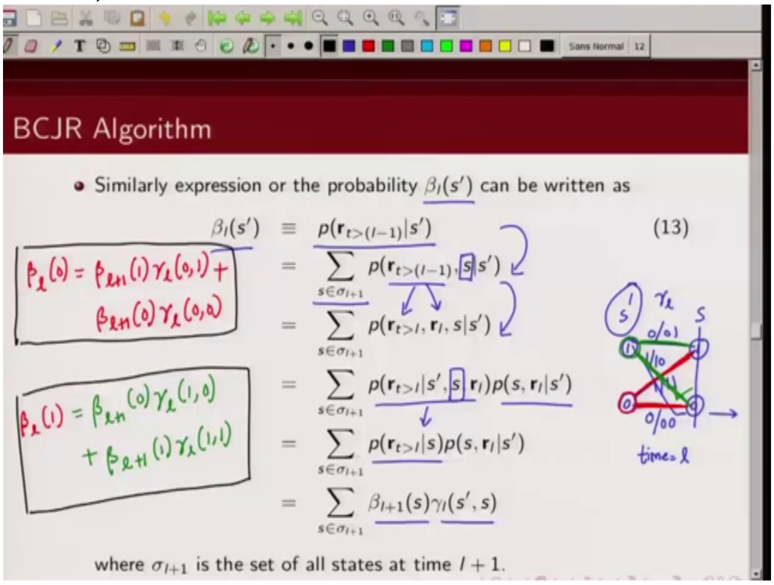
So this is our expression for betas. So as you can see similar to the expression for alphas now these betas

(Refer Slide Time 33:09)



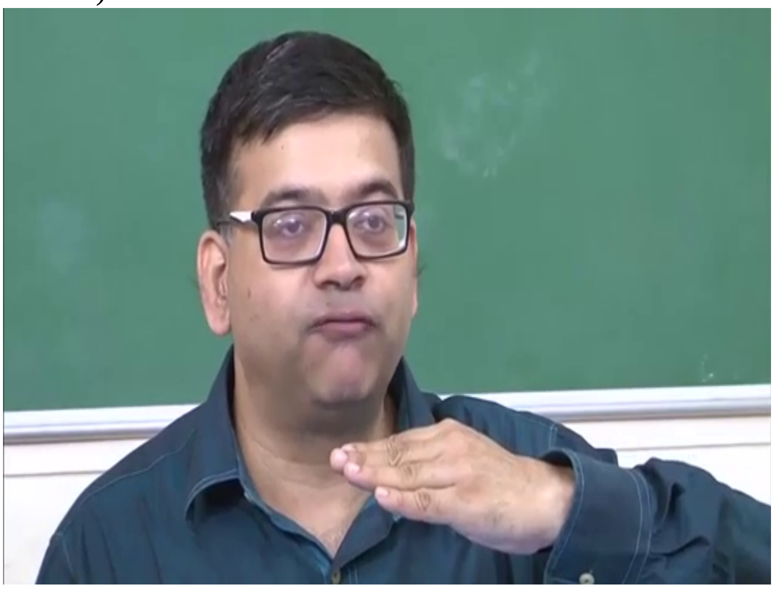
can be computed using,

(Refer Slide Time 33:13)

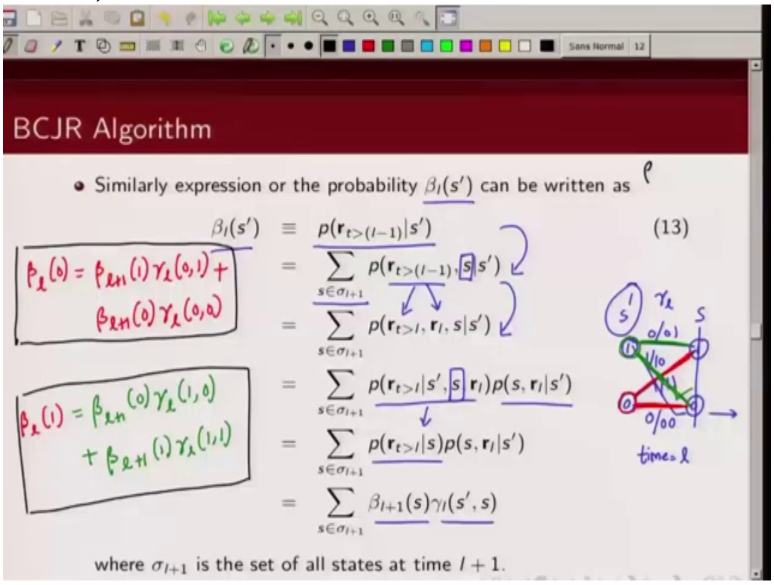


so alphas can be computed using

(Refer Slide Time 33:15)

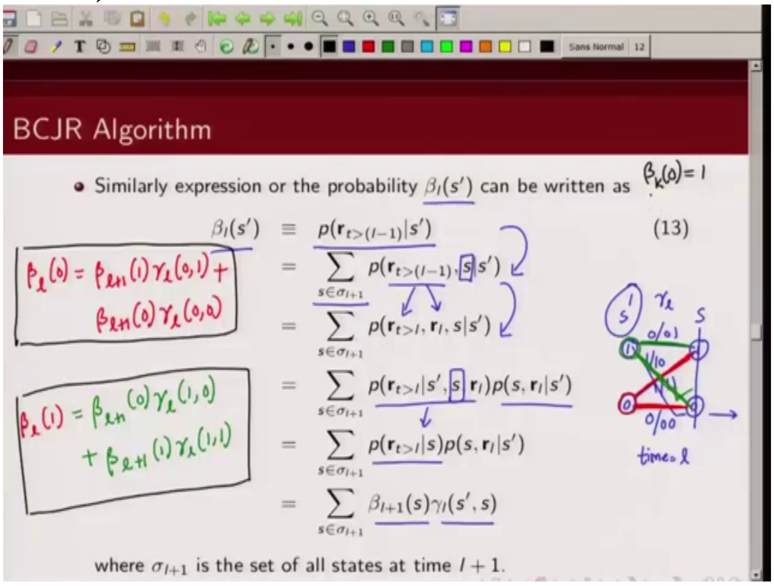

forward recursion and similarly betas can be computed using backward recursion. So then we would require the knowledge of beta at time at end of the Trellis. Now how do we know the values of beta? Now if the encoder is terminated, that means if the encoder is brought back to all zero state in that case, beta at end of Trellis, at end of the time, let's call beta at time k

(Refer Slide Time 33:51)



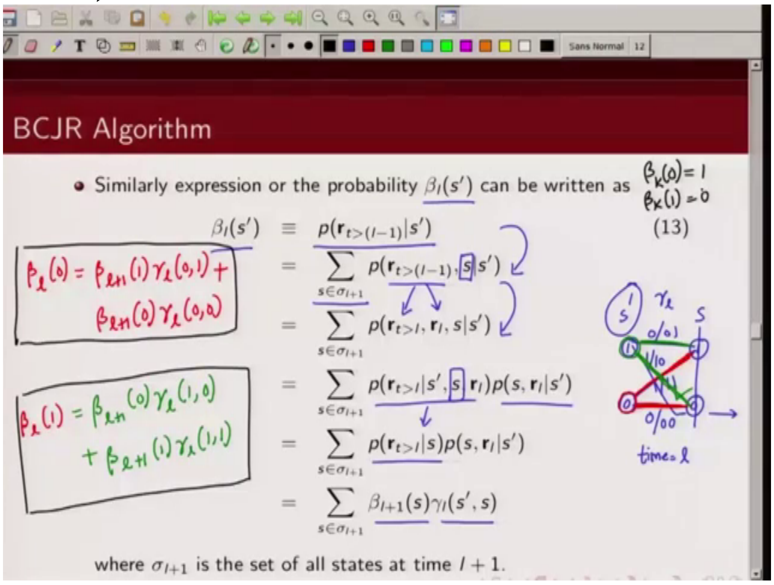
which is the end of the block, at state 0 will be 1 and for all other state

(Refer Slide Time 33:57)



it will be, in this case there are only 2 states, so for all other states it will be 0.

(Refer Slide Time 34:04)

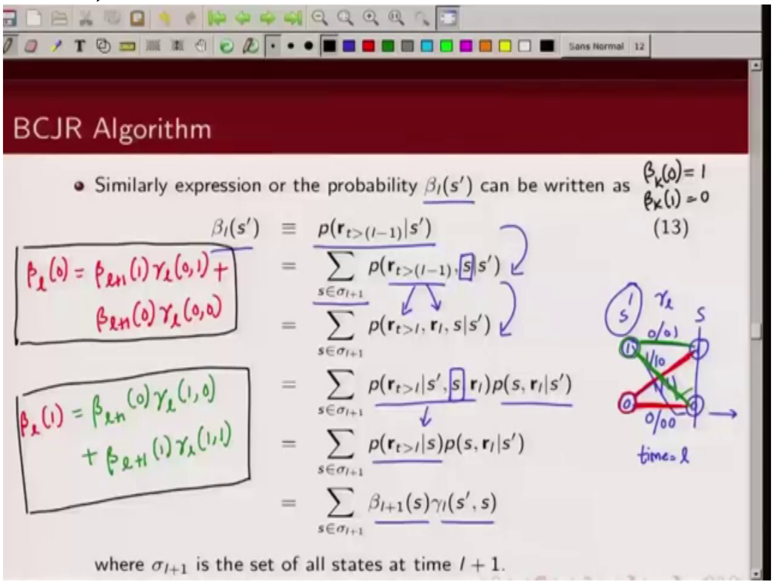


This is for the case when the convolutional encoder is brought back to all zero

(Refer Slide Time 34:11)

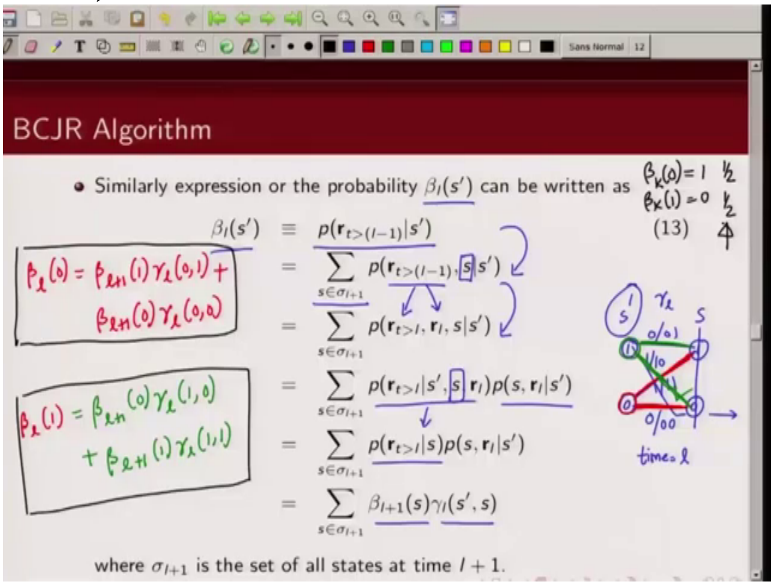
state, it is terminated. In case the convolutional encoder is not terminated, then we don't know in which state it has uh ended up with. So what we will do is in that case we will assume

(Refer Slide Time 34:38)



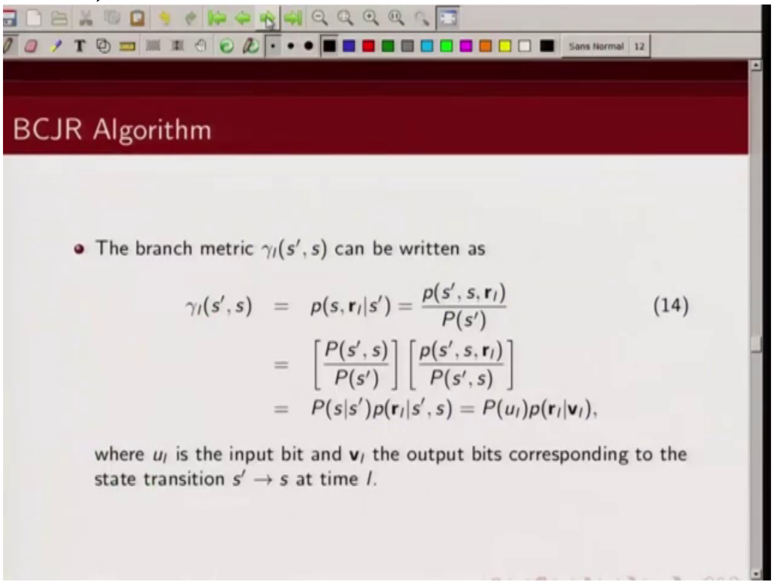
that it is equally likely to end up at all zero state or any other state. So in that case, we would assume beta at the end of the block to be equal to 1 by number of states. So in this case, we would assume that beta k 0 is half and beta k 1 is half. So this is for the case when convolutional encoder is not

(Refer Slide Time 34:56)



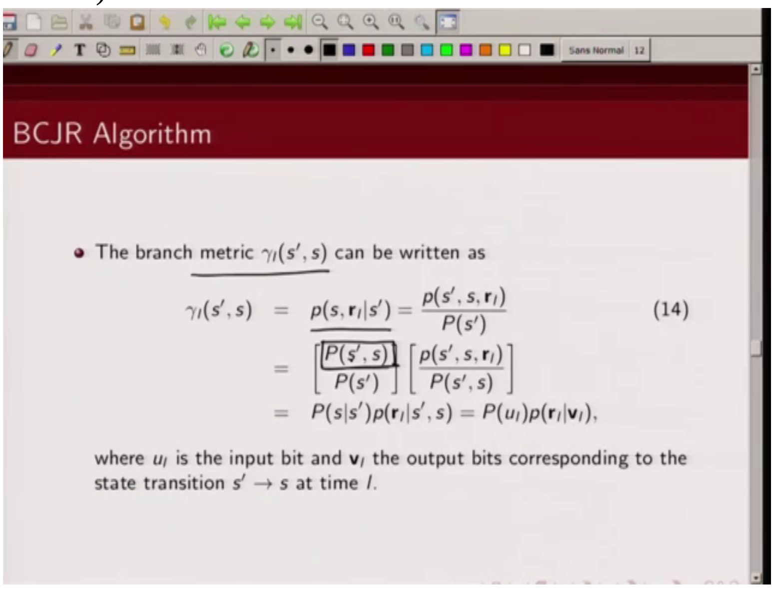
terminated. That means it is not brought back to all zero state and this will be the initial condition when the convolutional encoder is terminated.

(Refer Slide Time 35:07)



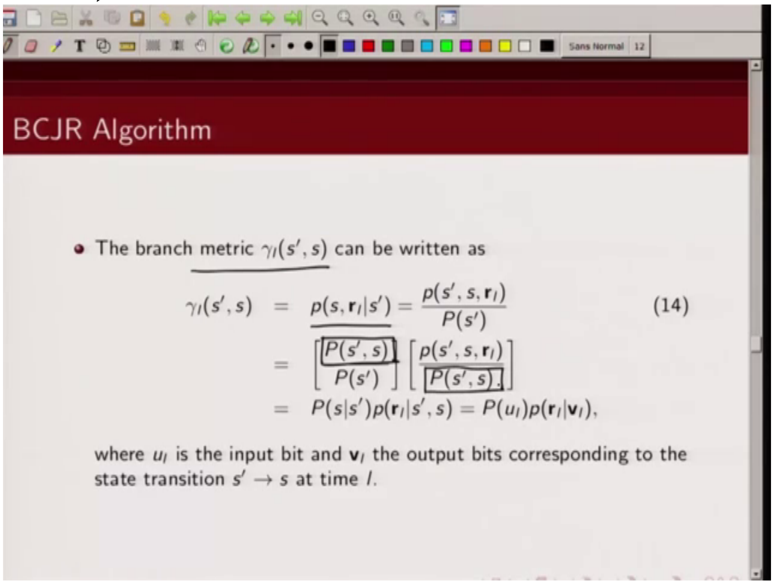
Now next we compute the branch metric gamma. Now from definition gammas can be written like this. So this can be written as joint probabilities of being in previous state s prime next state s given a received sequence at time l r l divided by probability of being in previous state s prime. Now this I introduce the term this so I

(Refer Slide Time 35:40)



add this term in the numerator, similarly I add this term in the denominator, Ok.

(Refer Slide Time 35:46)

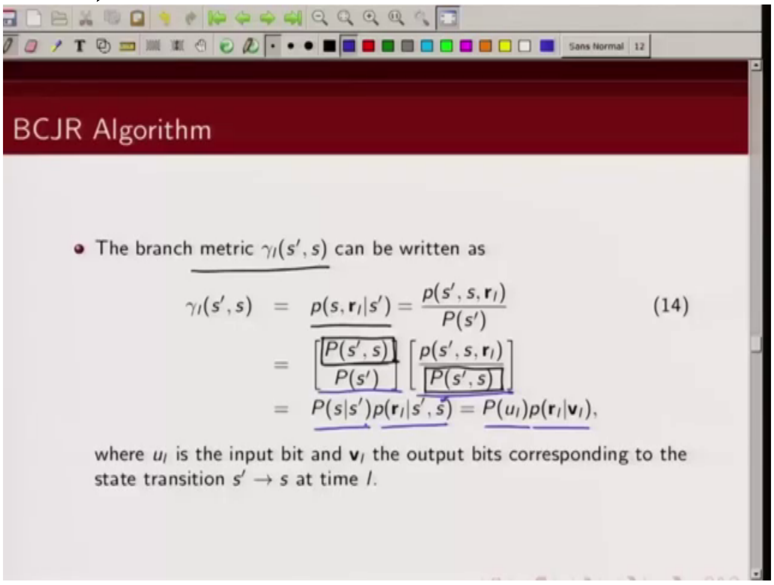


Now this, this quantity can be written as probability of s given s prime and this probability can be written as probability of r l given previous state s prime and next state s which can also be written as probability of r l given transmitted sequence v l multiplied by a priori probability of getting u l. So note that this probability will be 1 only when there is a valid transition from

(Refer Slide Time 36:22)

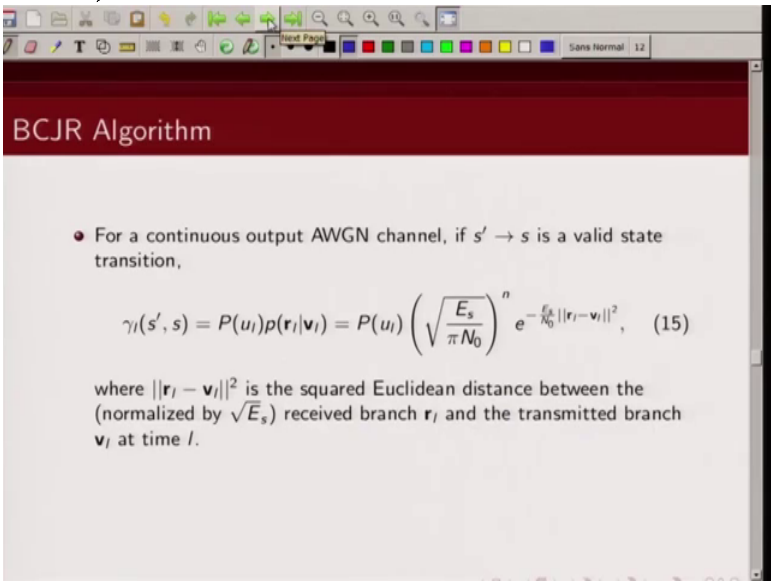
state s prime to s, otherwise this will be 0, Ok.

(Refer Slide Time 36:30)



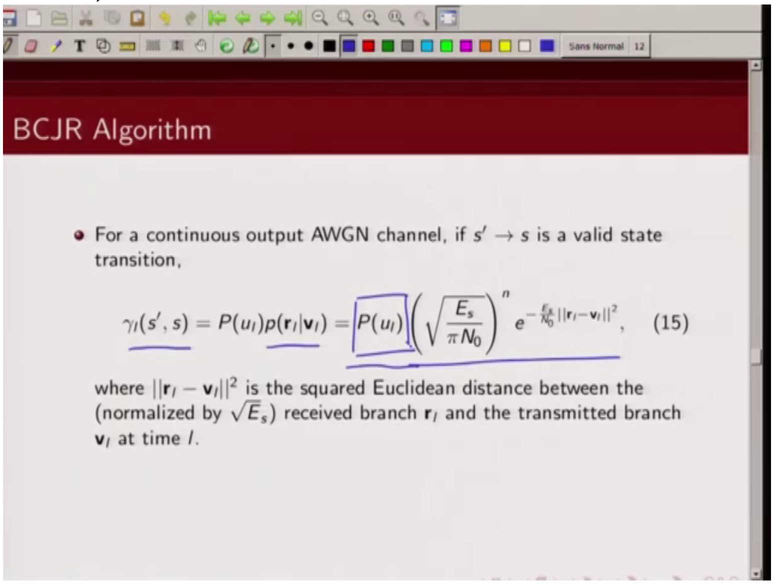
So what does gamma depends on, it depends on what is the a priori probability of u l and it depends on this likelihood function, probability of r l given v n.

(Refer Slide Time 36:46)



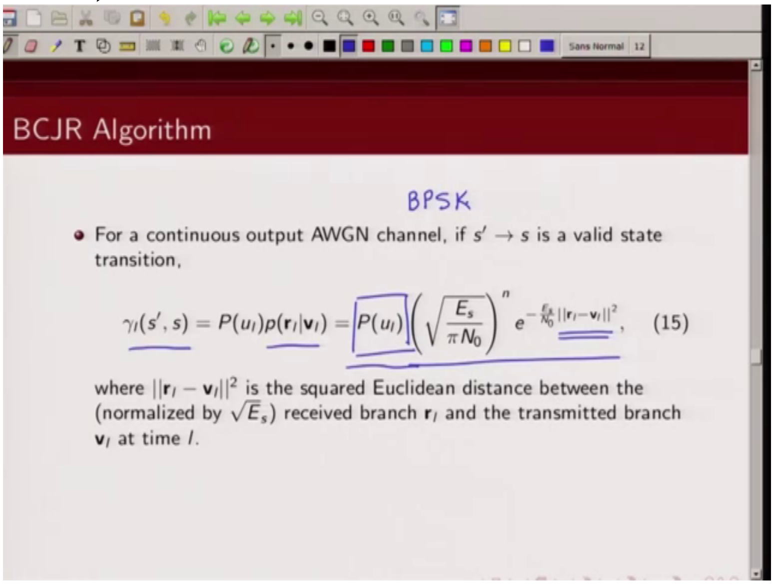
Now if we consider an additive white gaussian noise channel, we can write this probability of r l given v in this particular fashion. So gamma for an a w g n channel will then be given by this expression. So note this depends on a priori

(Refer Slide Time 37:09)



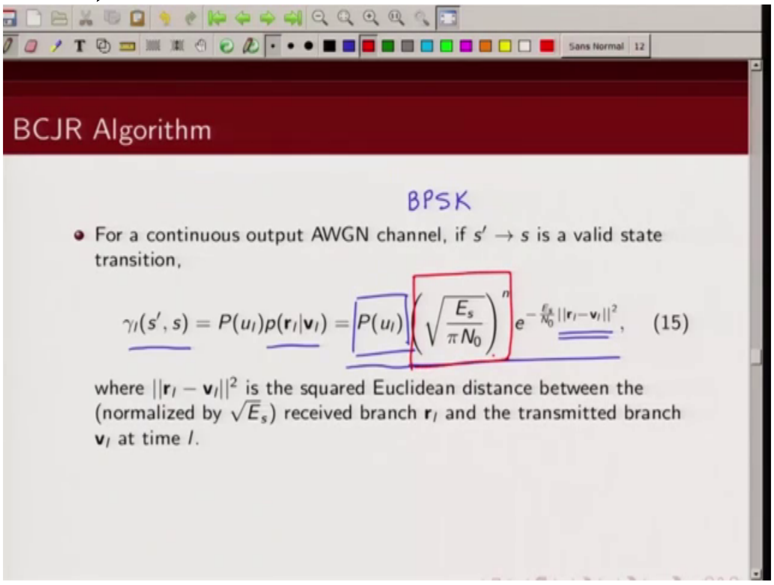
probability of u l. It depends on the Euclidean distance between r l and v l. Now let us assume that we are considering a binary phase shift keying. So in other words basically we have bits mapped to plus 1 and minus 1 let's say or plus

(Refer Slide Time 37:31)



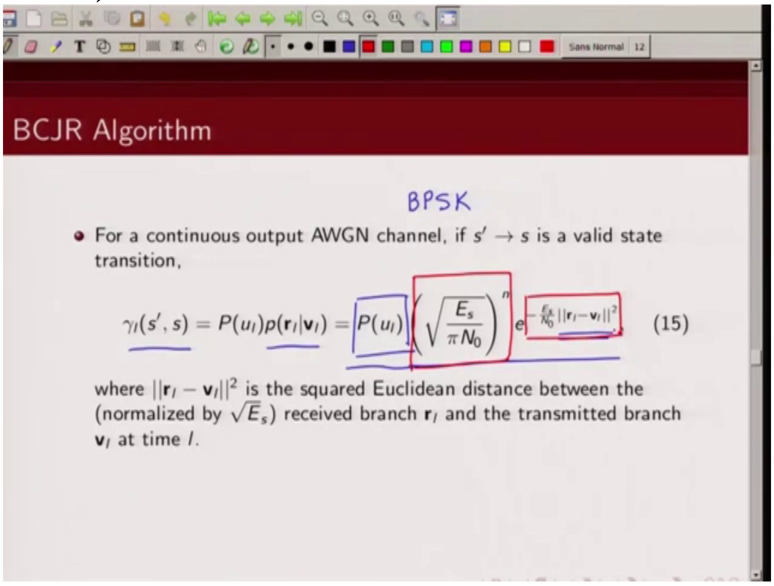
E s and minus E s, Ok. So let us expand this term and see can we simplify this term? Now this term is, this term will be common for all the terms which is, which depends only on

(Refer Slide Time 37:49)



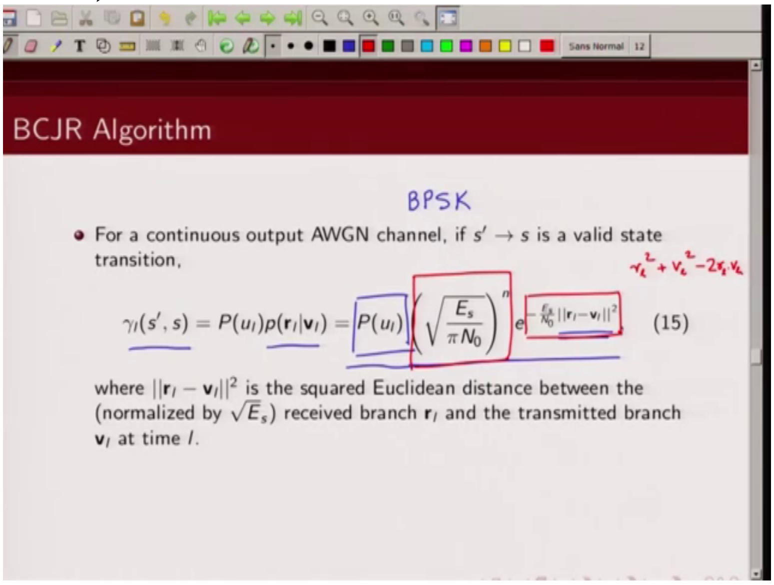
signal to noise ratio. And if you look at this particular term, so here there is a

(Refer Slide Time 37:56)



r l square term, there is a v l square term and then there is minus 2 r l v l term. So this r l,

(Refer Slide Time 38:09)

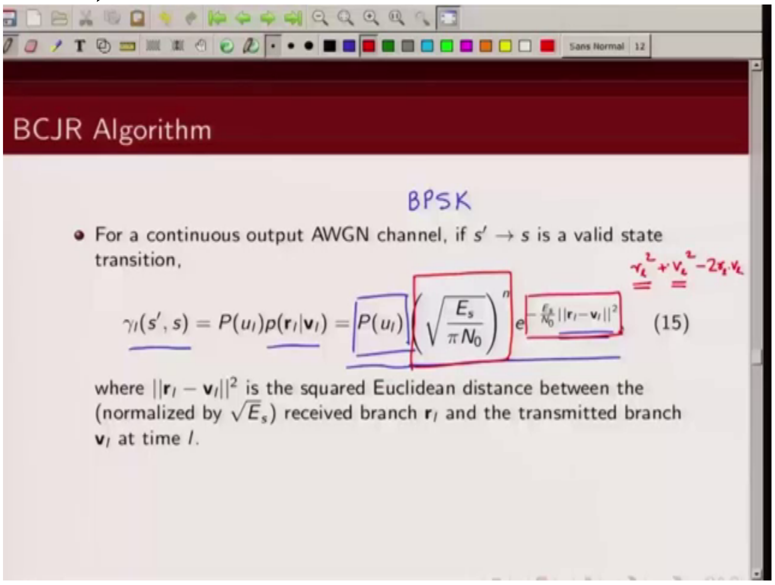


r l square term that does not depend on what my v l is. And since we are considering a b p s k modulated signal, so v l whether u of l is

(Refer Slide Time 38:23)

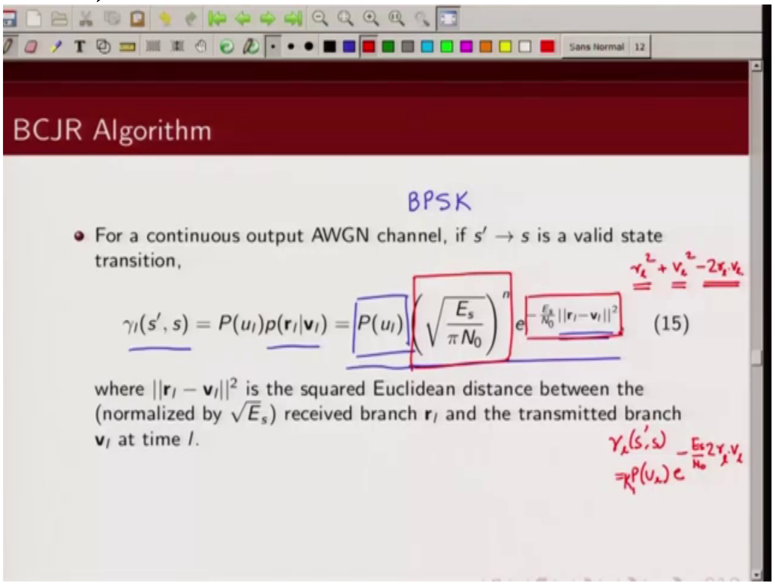
minus 1 or plus 1, this will basically be the same. This will be just 1.

(Refer Slide Time 38:28)



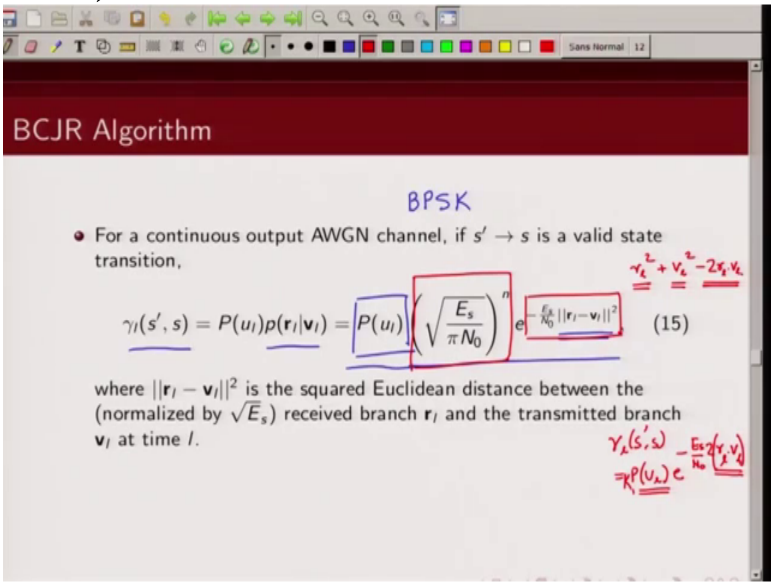
So the only term that is changing with choice of v l is this particular term. So what we can simplify this gamma l, we can just simplify our gamma l like this. So this is basically probability of u l and exponential minus E s by n naught and this is basically two times r l v n so it becomes dot product between the recieved sequence and uh this transmitted codeword v l. So, and of course there is some constant, there is some constant term k 1

(Refer Slide Time 39:12)



which is common. So in nutshell then, our gamma depends on this term, right and it depends on what the initial a priori probability of u l is. So for an additive white gaussian noise channel when we are applying b p s k modulation then we can simplify our expression for gamma. So this can be written as E raised to power minus E s by N naught by 2 times r l dot v l.

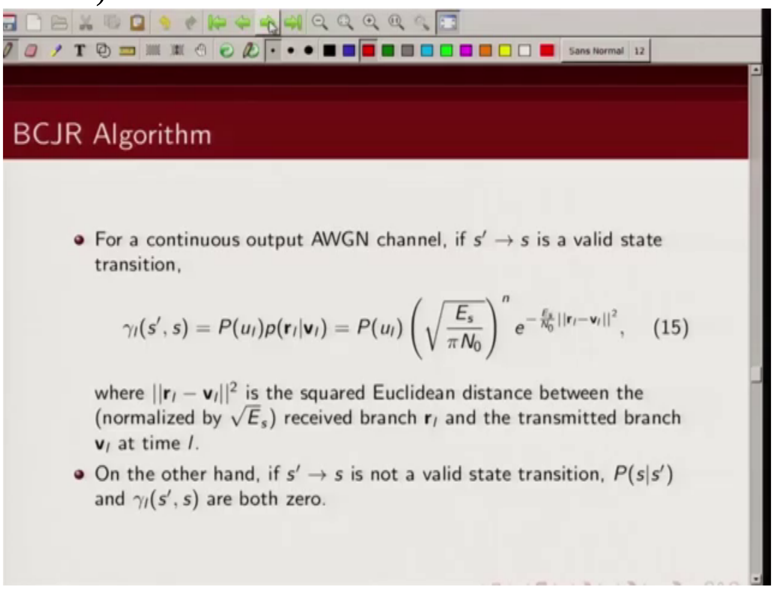
(Refer Slide Time 39:49)



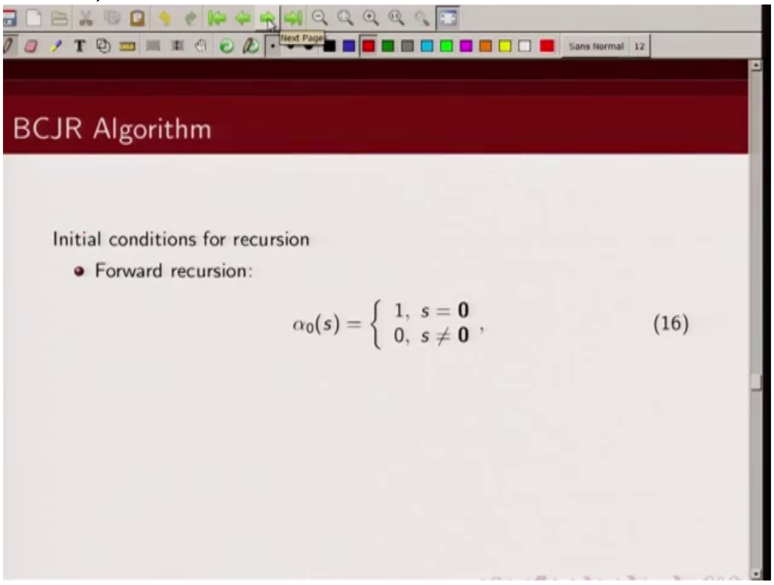
Now what is this r l dot v l? We will illustrate this with an example when we solve, when we show an example Ok. So the point which I am trying to make is that this expression that you see for computation of gamma for additive white gaussian noise, it essentially depends on two terms. One is this, and another is this term.

Next

(Refer Slide Time 40:16)



(Refer Slide Time 40:17)

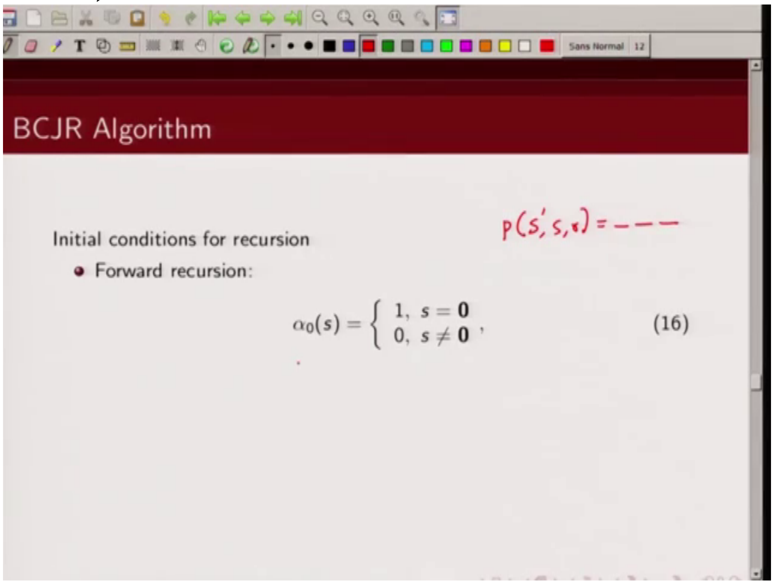


I have already specified now that

(Refer Slide Time 40:22)

our joint probability of, the joint probability that we computed, it's basically a product of 3 terms, alpha, beta and gamma. Now alpha beta can be computed in a recursive fashion. And I already mentioned that

(Refer Slide Time 40:38)

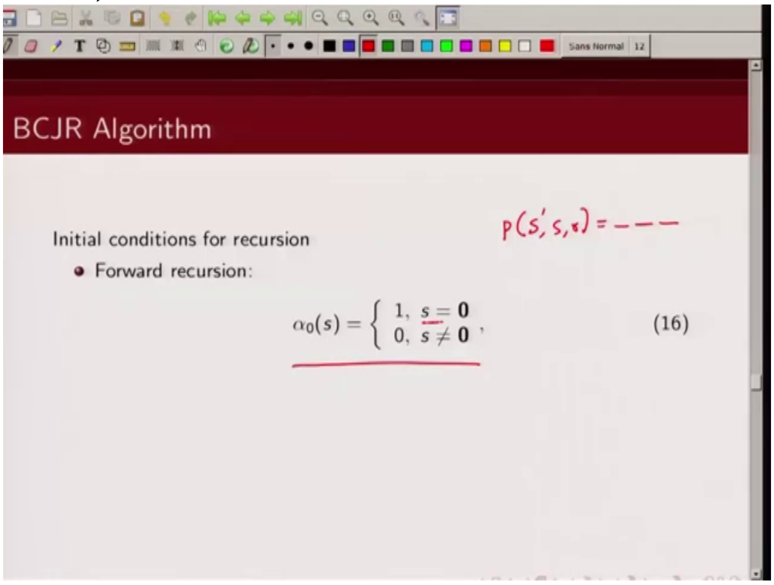


usually our encoder is in all zero state to start off with

(Refer Slide Time 40:43)

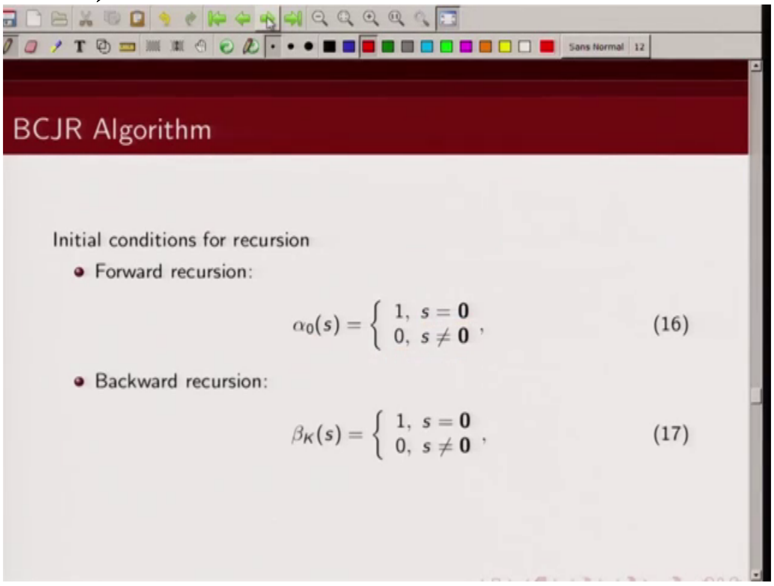
and that's why we assume that

(Refer Slide Time 40:46)



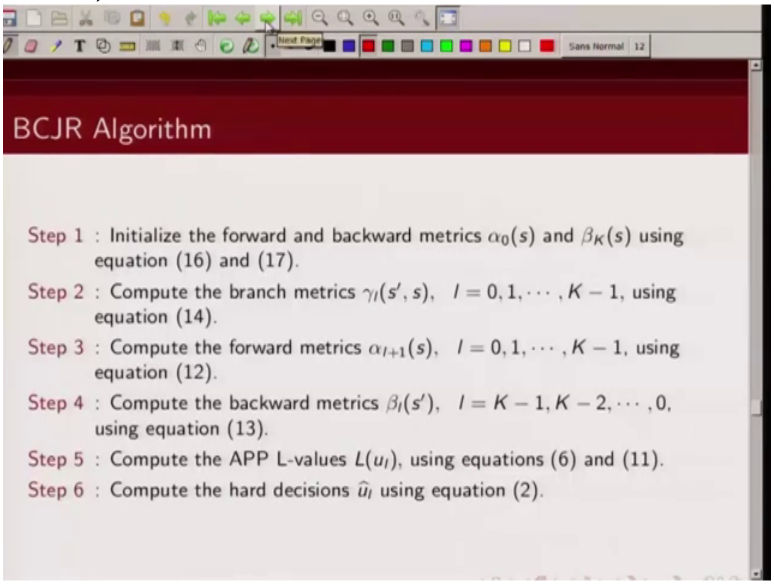
alpha times 0 is 1 for the state 0 and it is 0 for all other states. Similarly

(Refer Slide Time 40:59)



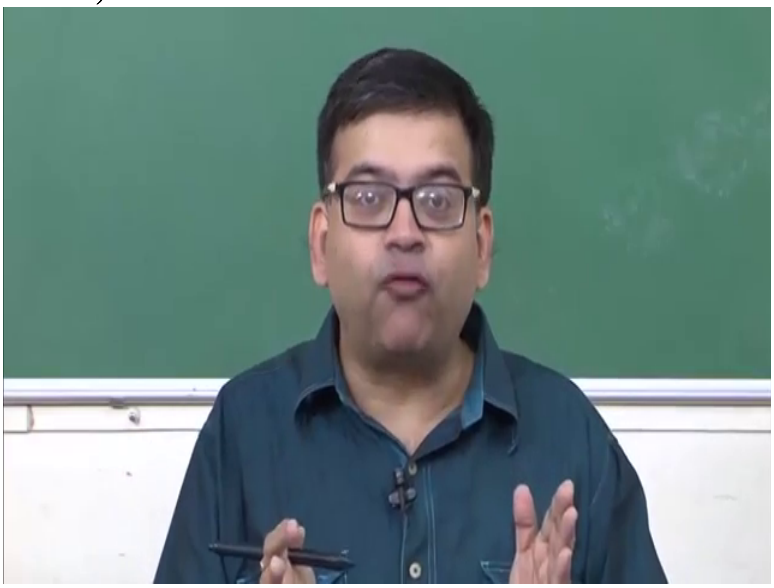
if we assume that our encoder is terminated, that means it has been brought back to all zero state in that case at the end of our block which is our k, beta k will be 1 for state 0 and 0 for all other states. So these are our initial conditions for computing the recursion for, for forward recursion as well as backward recursion. So now then,

(Refer Slide Time 41:31)



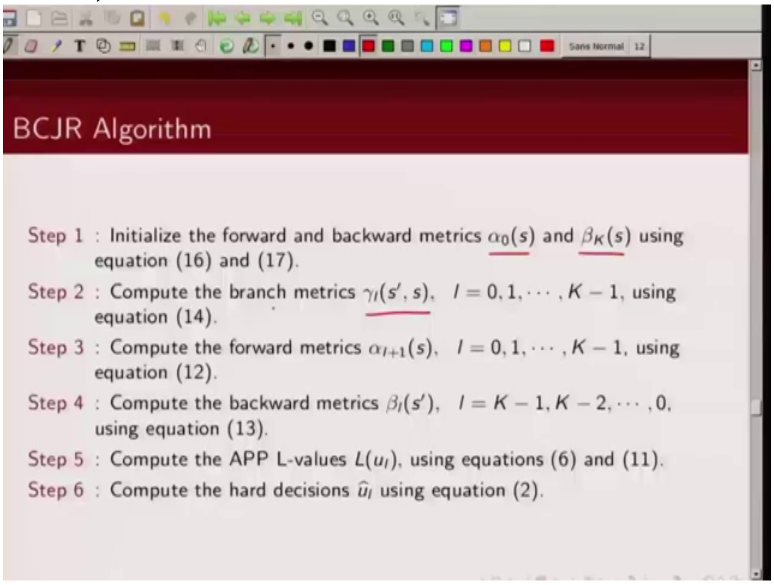
to recap how do we compute the a posteriori probability. The first thing is we need to initialize the values of alpha times 0 and beta times end of the block which is I am calling k. The next thing I need to do is, now to compute alpha and beta I need the value of this branch metric gamma. So the first thing I need to do is I need to compute this branch metric gamma. So I will compute this branch metric for all

(Refer Slide Time 42:06)



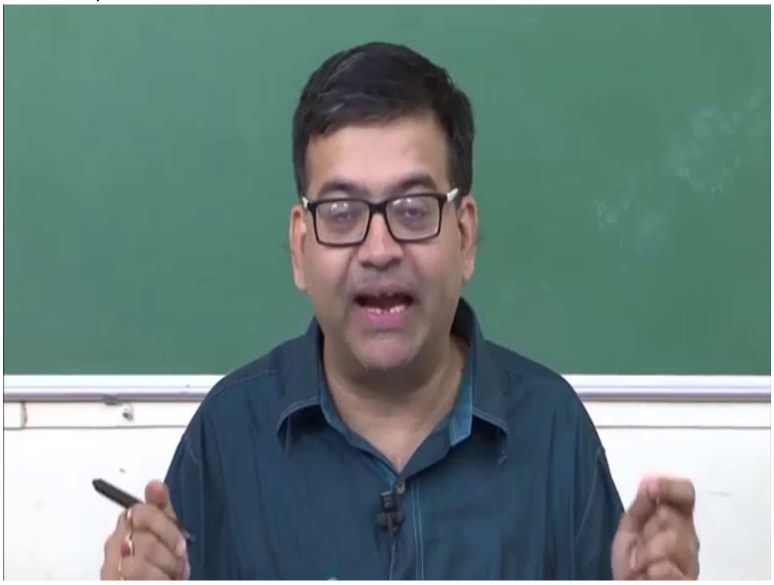
valid transitions and for all time instances. So that's the second step.

(Refer Slide Time 42:13)



The third step is once I compute this branch metric gamma then I will compute using forward recursion, I will compute the values of alphas and using backward recursion I will compute the values of beta. Once I have the values of alpha, beta and gamma

(Refer Slide Time 42:34)

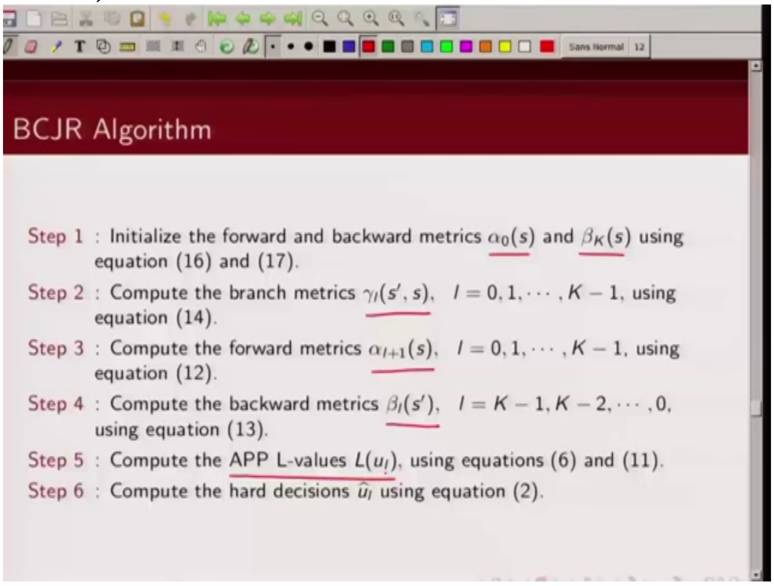


then I can compute the a posteriori probability because I

(Refer Slide Time 42:40)

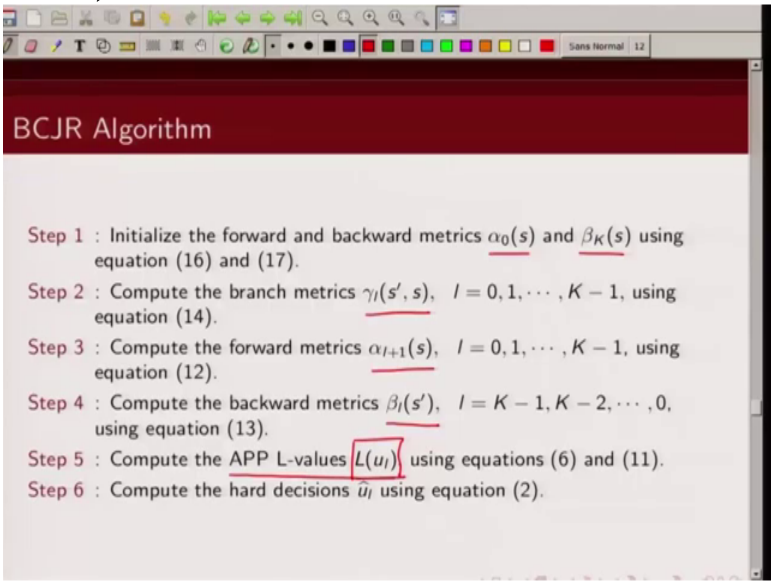
have shown that it is a basically product of these three terms. So I can then compute

(Refer Slide Time 42:46)



these A P P values and once I have these A P P

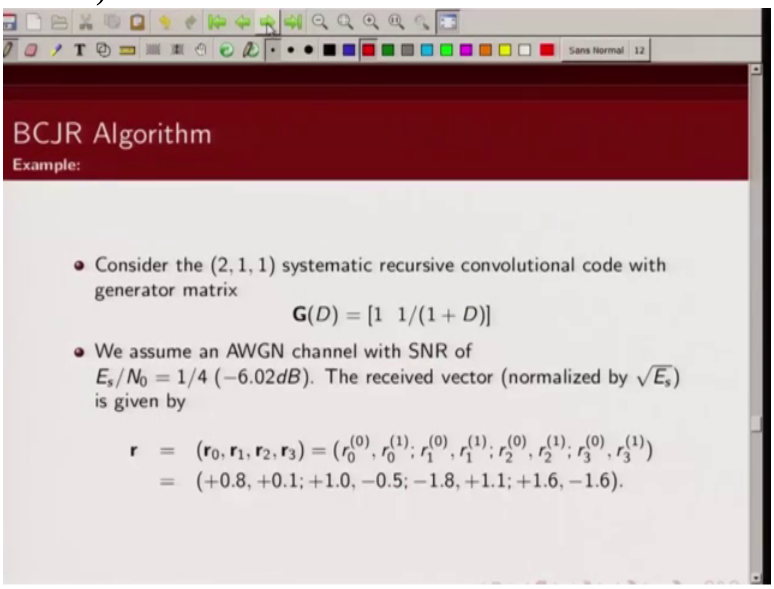
(Refer Slide Time 42:52)



value I will take a hard decision based on whether this is greater than 0 or plus 1. So the final thing that I am going to do is I am going to take a hard decision based on what is the value of this A P P value, Ok.

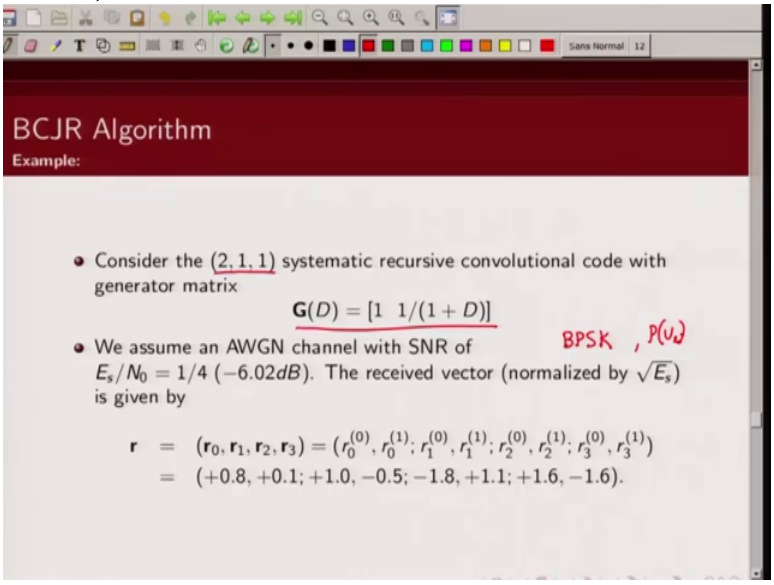
So let's now

(Refer Slide Time 43:14)



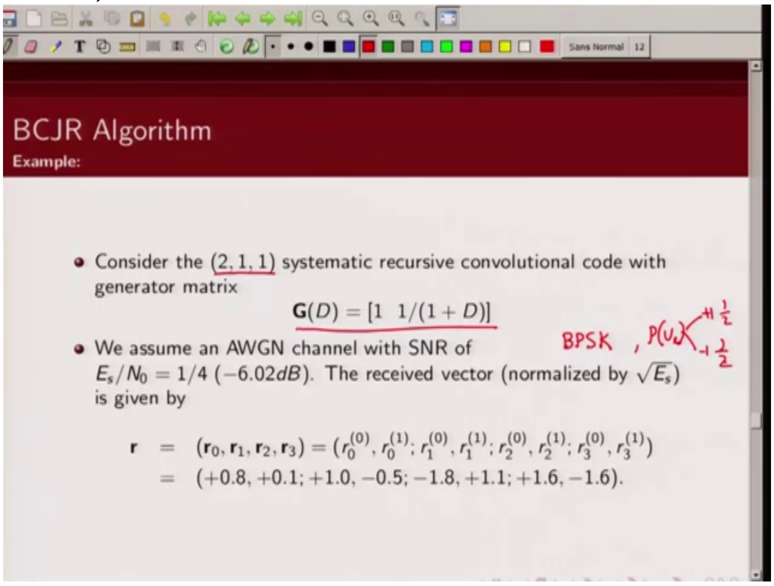
show the same using an example. So we are going to consider an example to illustrate how we can do b c j r decoding. So we are considering a rate 1 by 2 convolutional code with memory 1 whose generator sequence is basically given by this. The generator matrix is given by this. We are considering b p s k modulation and we are assuming that initial probability u l is equally likely to be plus 1

(Refer Slide Time 43:49)



or minus 1. So we are assuming it is equally likely to be, it is plus 1 with probability half and minus 1 with probability half. We are

(Refer Slide Time 43:58)

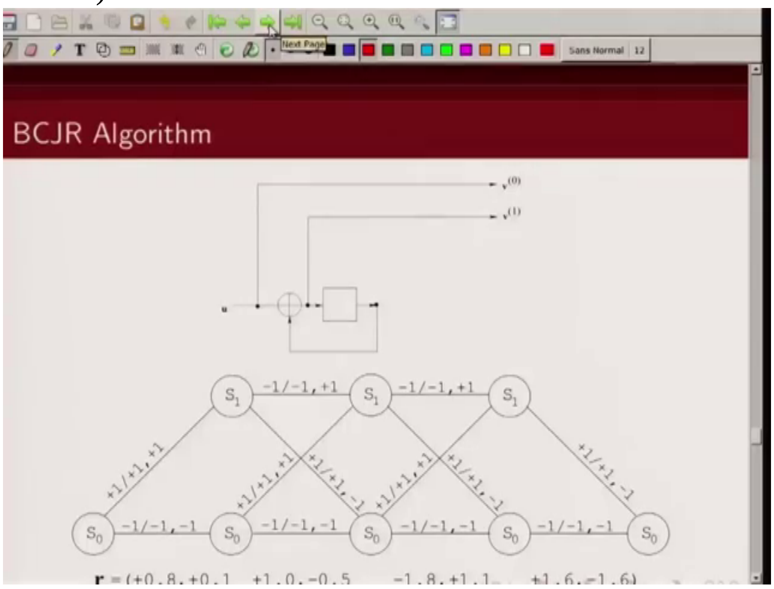


considering an a w g n channel with s n r of 1 by 4 and we are assuming that recieved signal are normalized by under root E of s. So what we are receiving is this particular sequence. The question I am interested is if the recieved sequence is this, I am interested in estimating what was my information

(Refer Slide Time 44:24)

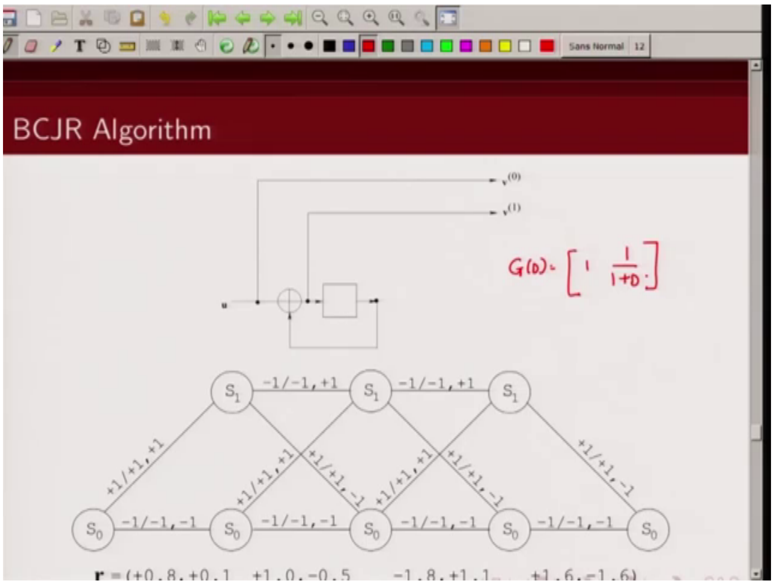
sequence. So to solve this problem what we need to do is we need to compute the a posteriori L value. Now to compute the a posteriori L value, we will first have to compute alpha, beta and gammas Ok and eventually we will compute the a posteriori L value and then we will take a hard decision on that to decide, estimate our information sequence. So this is the

(Refer Slide Time 44:59)



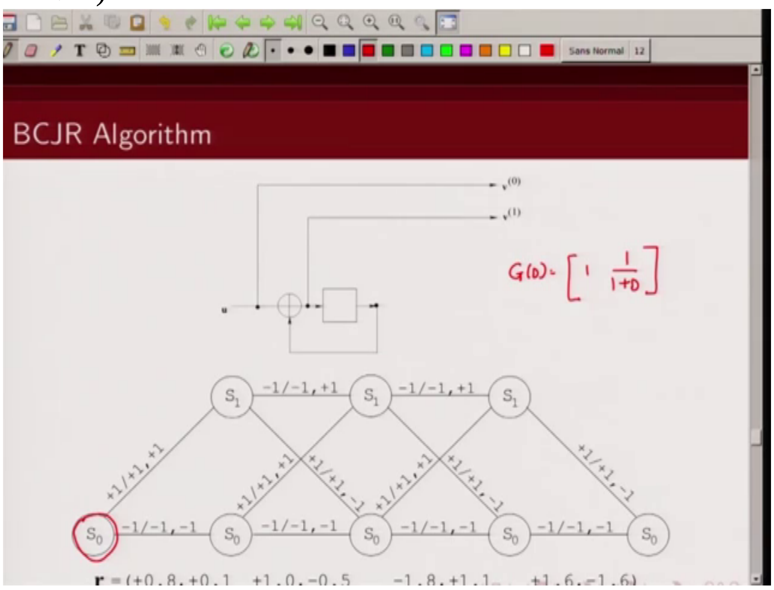
convolutional encoder that we have considered. This is basically G of D is rate 1 by 2 code and its

(Refer Slide Time 45:10)



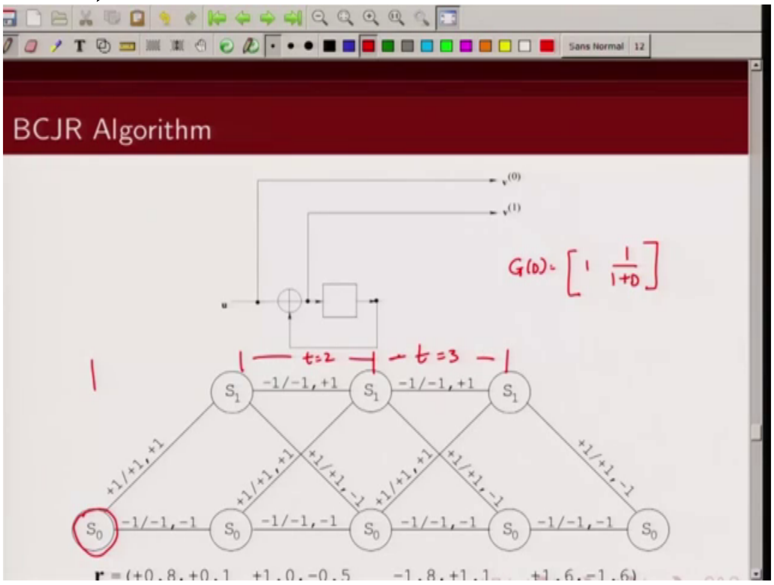
corresponding Trellis diagram is this. For simplicity I just considered 4 time instances. So initially I assume encoder is in all zero state

(Refer Slide Time 45:21)



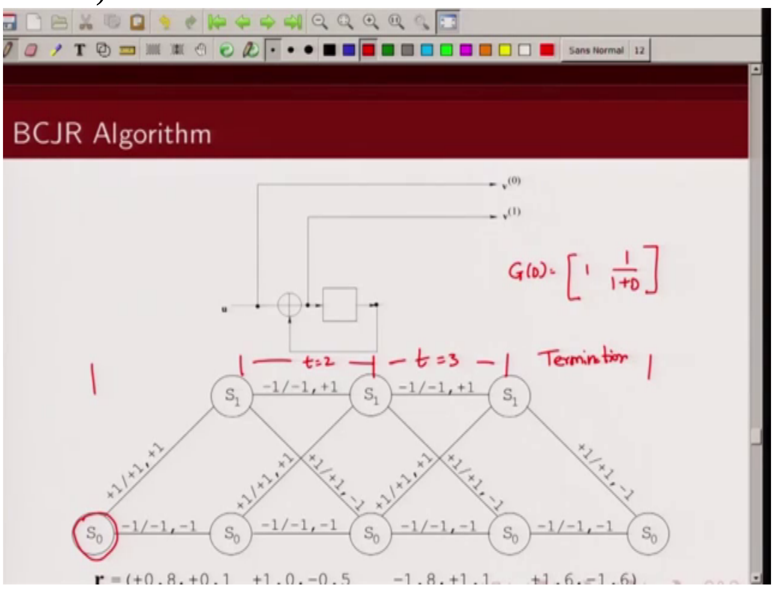
which is denoted by S 0 and it gets some bits. It moves to either S 0 or S 1 depending on what bits it recieved. This is first time instance; this is t equal to 2. This is, this is t equal to 3. And then after this what

(Refer Slide Time 45:42)



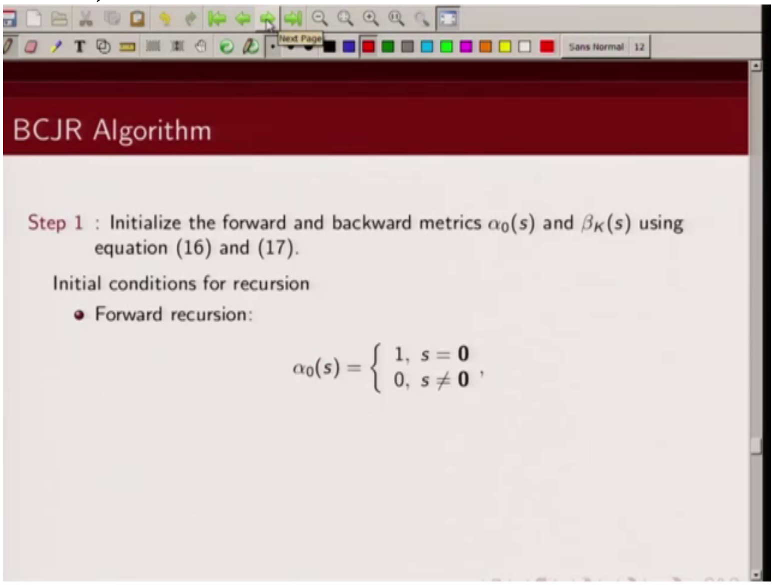
I am doing is I am terminating this encoder back to all zero state. So this is termination phase. So I bring this encoder

(Refer Slide Time 45:53)



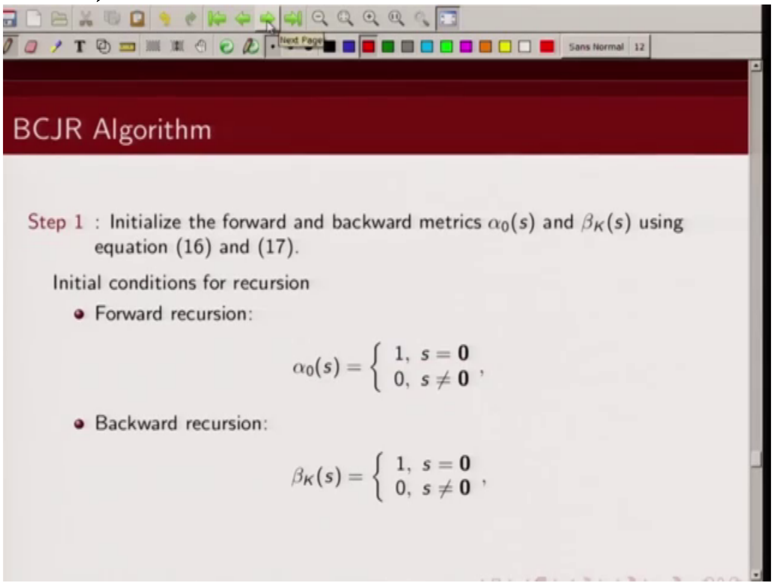
back to all zero state. Now this is a rate 1 by 2 codes, for each Trellis section I am receiving 2 bits. So at time t equal to 1, what I recieved is these 2 bits, point 8 and plus point 1. For t equal to 2, I recieved these 2, plus point 1 and minus point 5. For t equal to 3, I recieved minus 1 point 8 and plus 1 point 1 and for, during the termination phase I recieved plus 1 point 6 and minus 1 point 6. Please note I am interested in, given this recieved sequence I am interested in estimating what was the information bit that was transmitted at time t equal to 1. What was the information bit that was transmitted at time t equal to 2. What was the information bit that was transmitted at time t equal to 3? So

(Refer Slide Time 46:56)



as we said the first step was initializing alphas and betas for recursion; so since we started with all zero state, alpha at time zero for state 0 is 1, and for other states, which is state 1 it is zero. And since we are terminating

(Refer Slide Time 47:22)

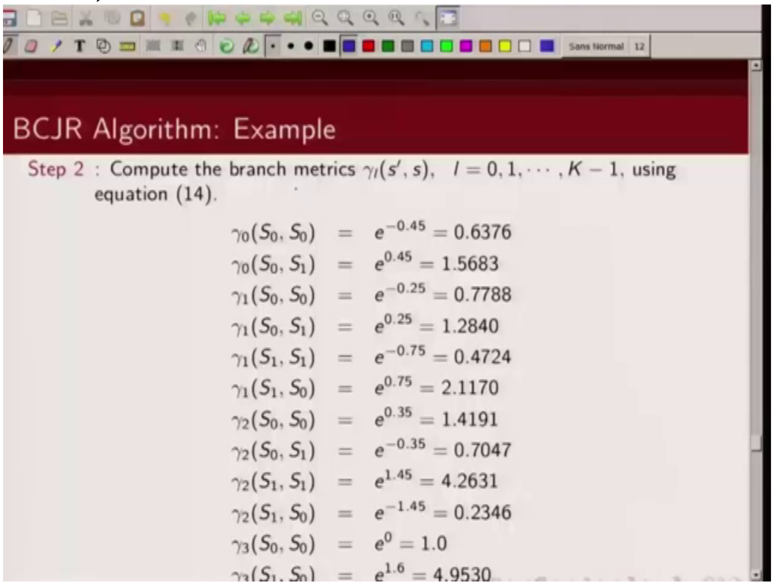


this encoder, so beta k times t equal to 4 is 1 for state 0 and it is 0 for other states which is state 1. So that's the first step. Initializing the forward and backward metric for time t equal to 0 and time t at end of the block, in our example t equal to 4. So once we have initialized our

(Refer Slide Time 47:52)

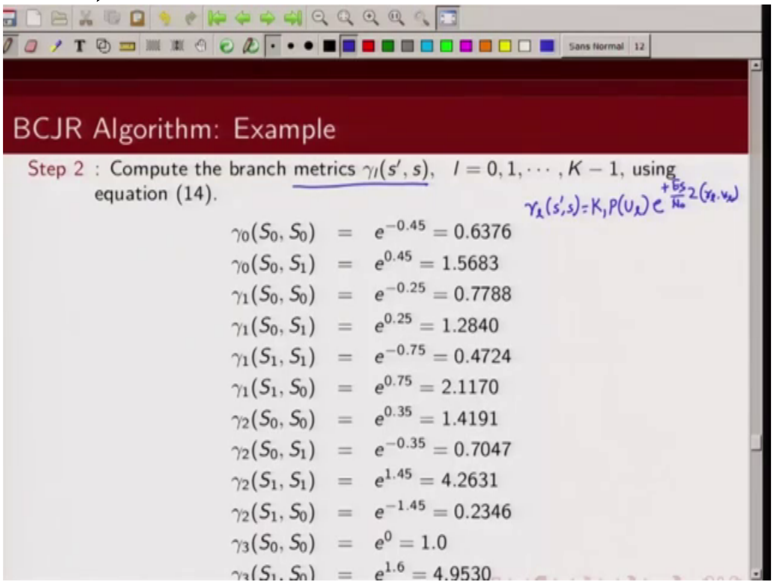
alphas and betas next we need to compute alphas and betas for other time instances and for that we would

(Refer Slide Time 48:04)



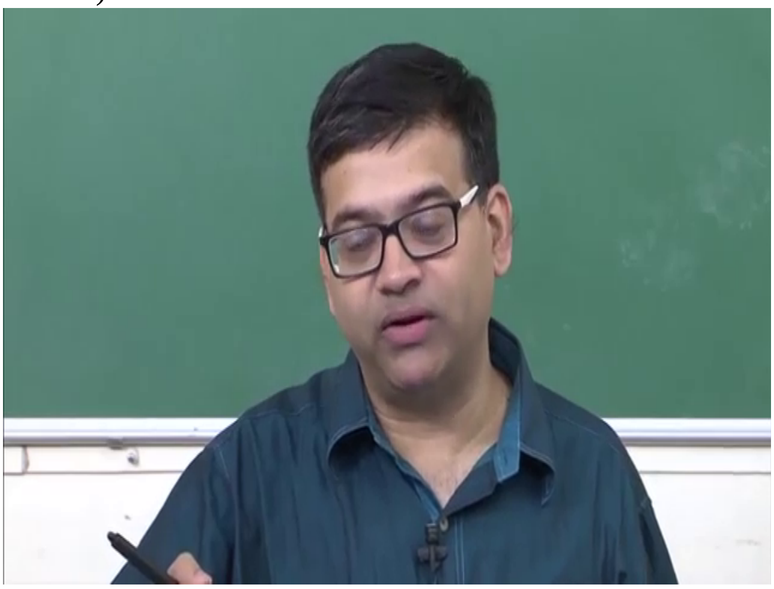
need our branch metric. Now how do we compute our branch metric? If you recall for the A W G N channel we showed that this branch metric can be written as some constant, say call it k 1 times probability initial a priori probability u l and we have exponential plus E s by N naught 2 times r l dot v l. Now in this particular example

(Refer Slide Time 48:33)



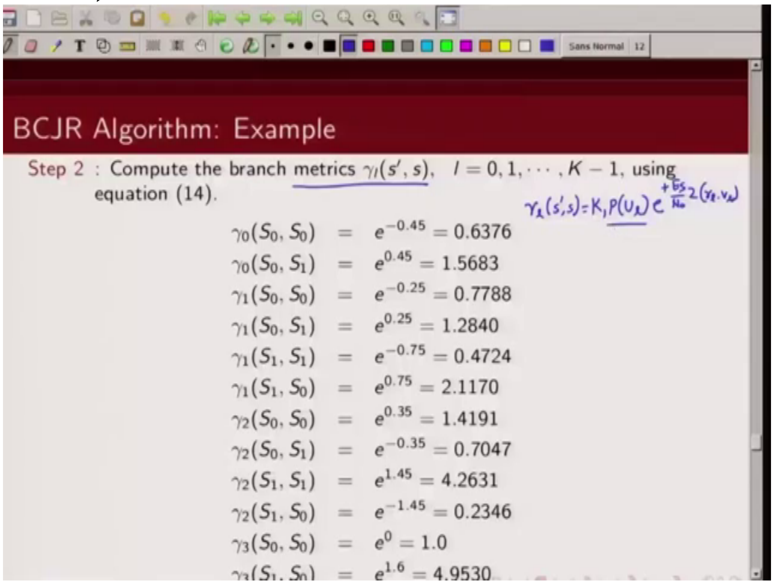
we are assuming that a priori it is equally likely to be plus one or minus one. So this

(Refer Slide Time 48:43)



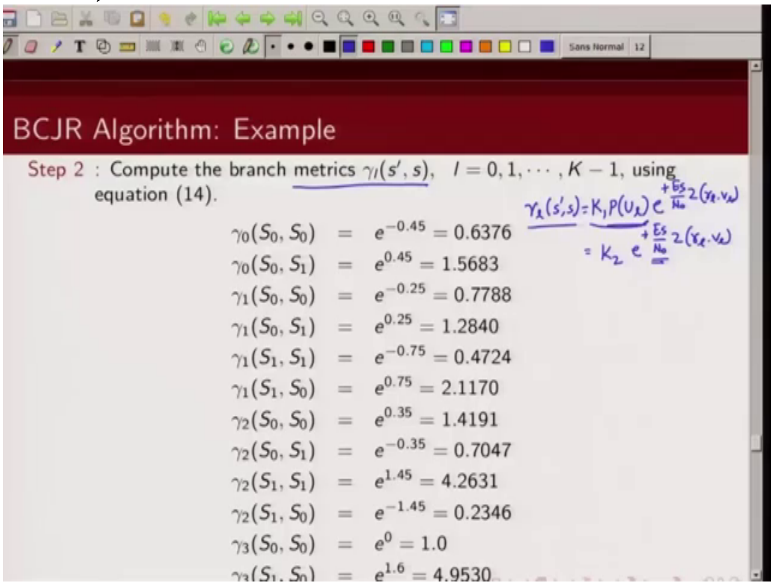
probability will be half whether u l is plus 1 or minus 1. So we can just, this will be

(Refer Slide Time 48:49)



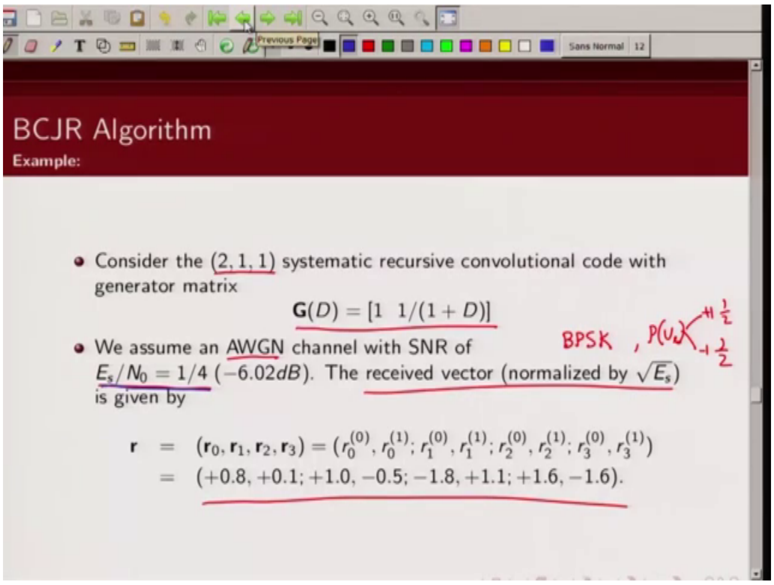
a constant, so we can just include this in a constant thing and we can just ignore this term. So what we need to compute, to compute the branch metric is basically some k 2 times exponential plus E s by N naught 2 times r l dot v l. Now in our example E s by N naught is 1 by 4.

(Refer Slide Time 49:14)



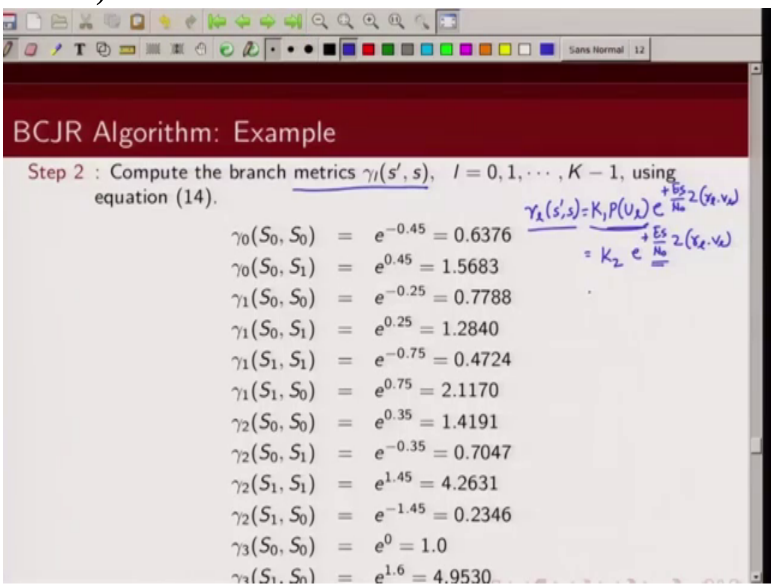
Just go back, E s by N naught is 1 by

(Refer Slide Time 49:19)



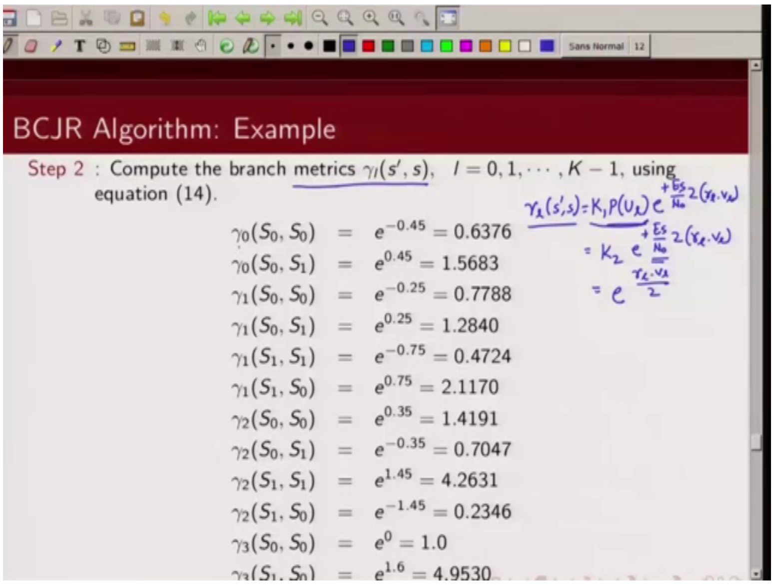
4. So then we can write this as,

(Refer Slide Time 49:24)



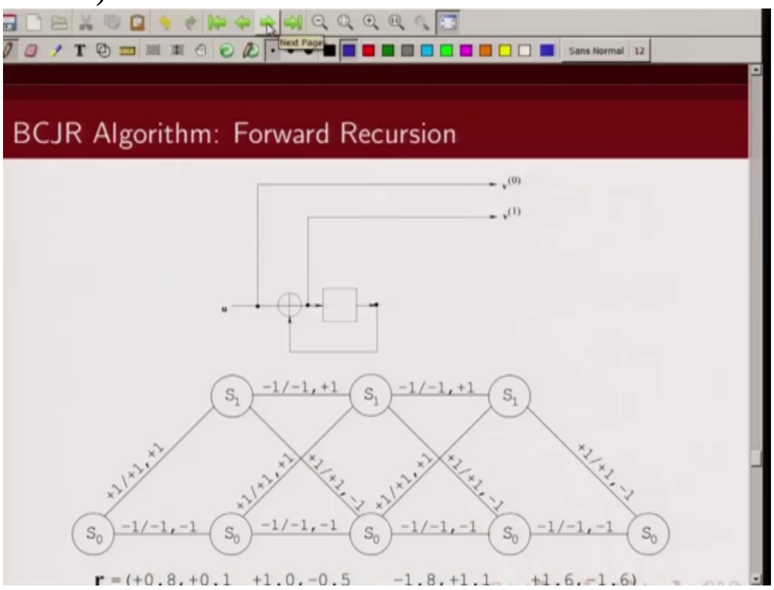
just ignore the constant term. I can write this as r l dot v l by 2. Now how do we compute r l dot v l? Let's take an example.

(Refer Slide Time 49:38)



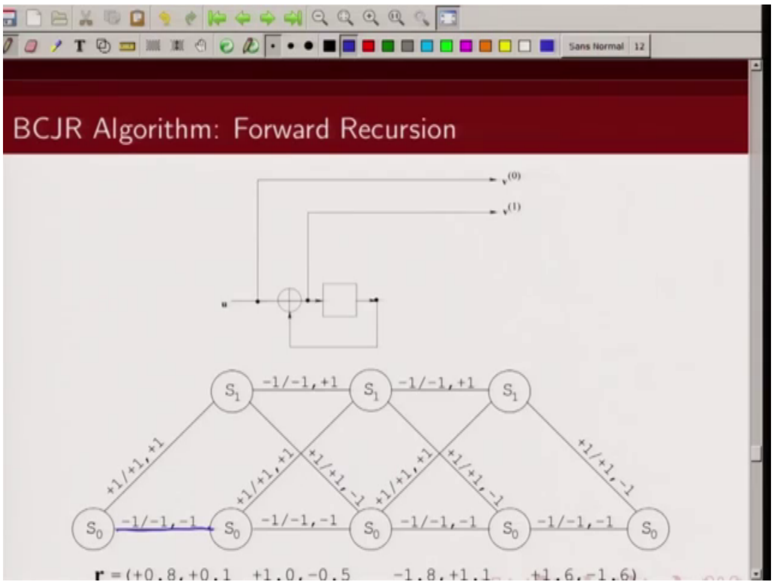
Let's take this. gamma at time t equal to 0, when the initial state is S 0 and final state is S 0, so what is this? This corresponds

(Refer Slide Time 49:52)



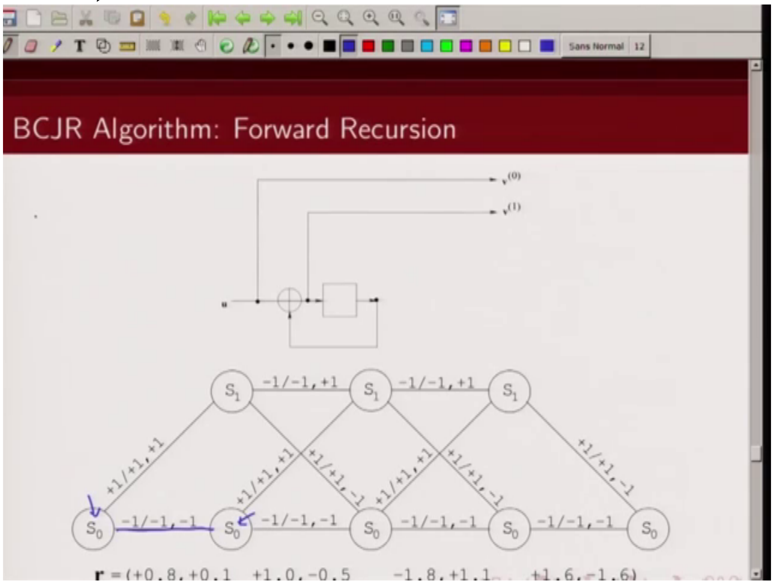
to branch metric for

(Refer Slide Time 49:55)



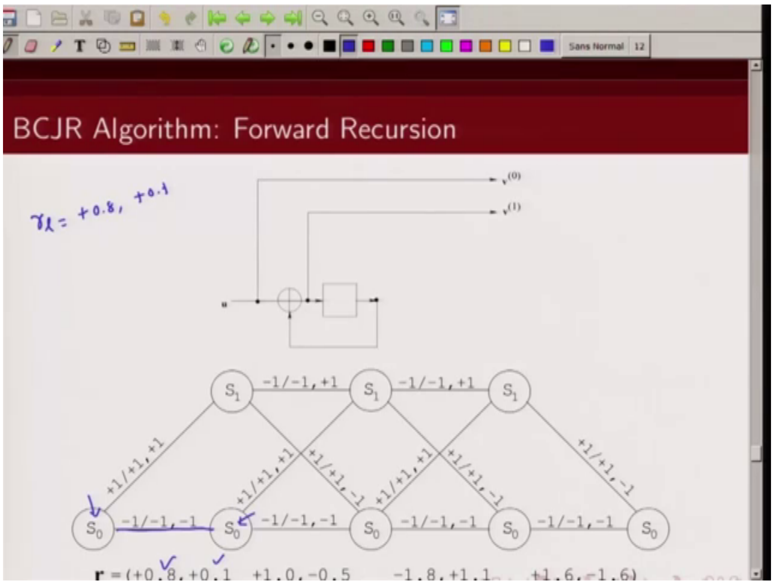
this path. This is time t equal to 0. Initial state is s 0, final state is s 0. Now what is

(Refer Slide Time 50:03)



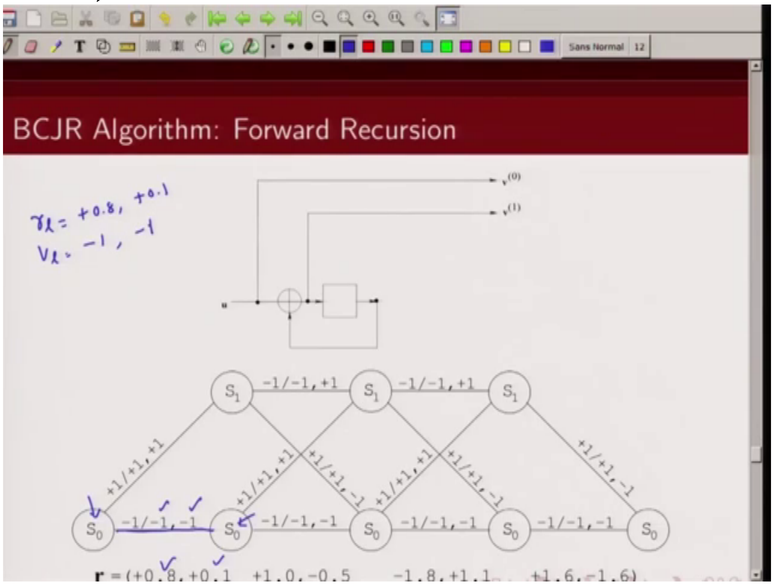
r L, what is the recieved sequence corresponding to this Trellis section? The recieved sequence is given by this. This is plus point 8 and plus point 1.

(Refer Slide Time 50:15)



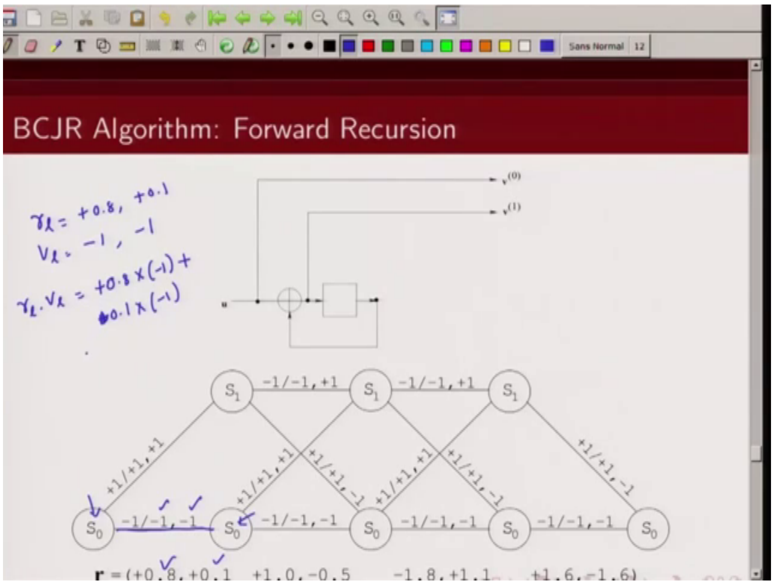
Now what is the v l corresponding to this transition? This is minus 1 and minus 1. So v l is minus 1 and minus

(Refer Slide Time 50:26)



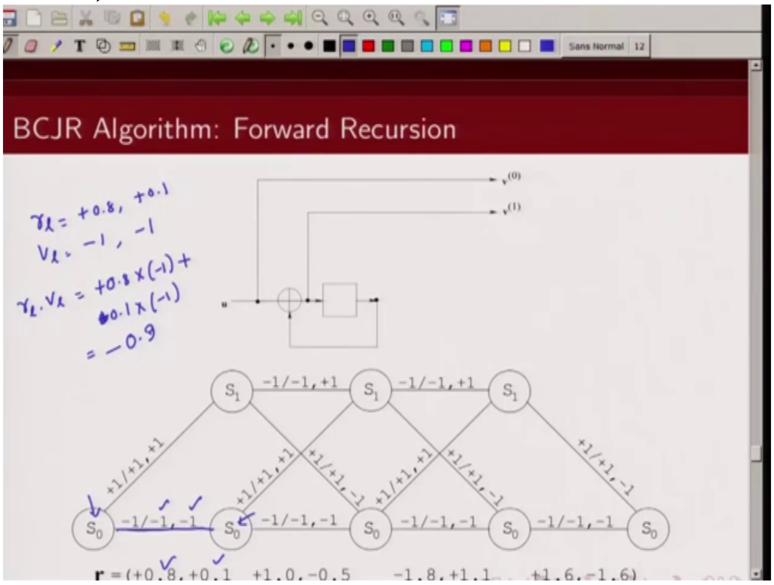
1. Then this dot product can be written as plus point 8 into minus 1 plus point 1 into minus 1. So

(Refer Slide Time 50:43)



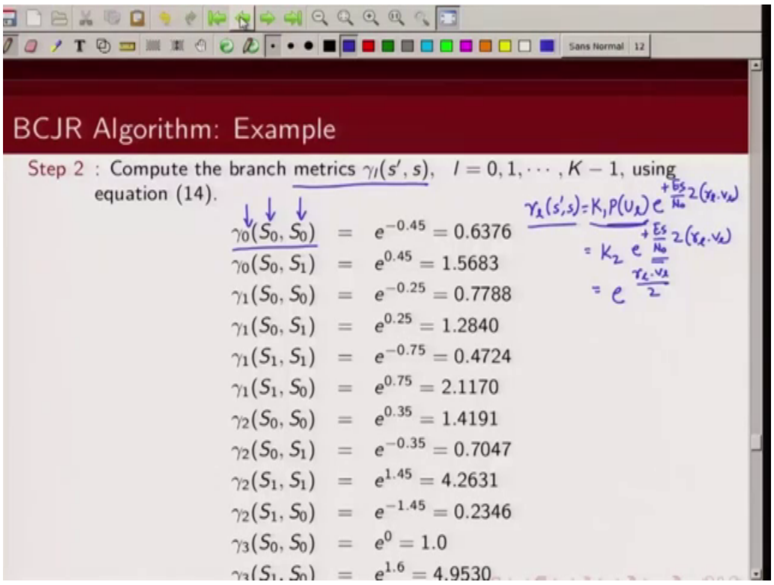
this will be minus point 9. So r l dot v l is minus point 9.

(Refer Slide Time 50:50)



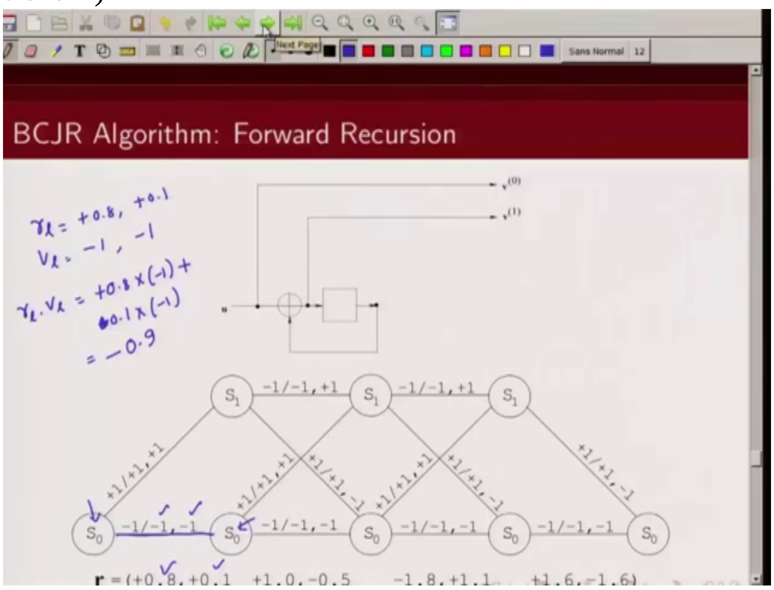
So if we plug that in here, minus point 9 by

(Refer Slide Time 50:54)



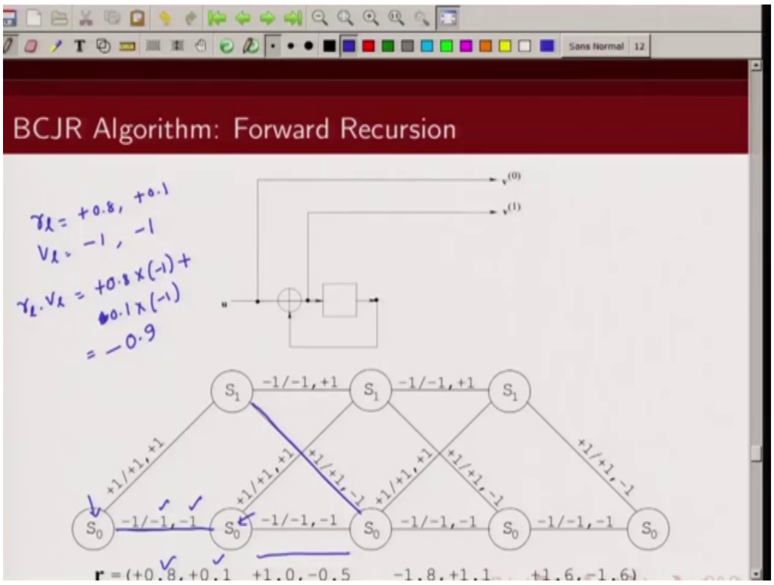
2, so this is minus point 4 5 which is given by this. Now you can take any other example. Let's just take this example. gamma 1 is S1 S 0, so what is this, so gamma 1

(Refer Slide Time 51:11)



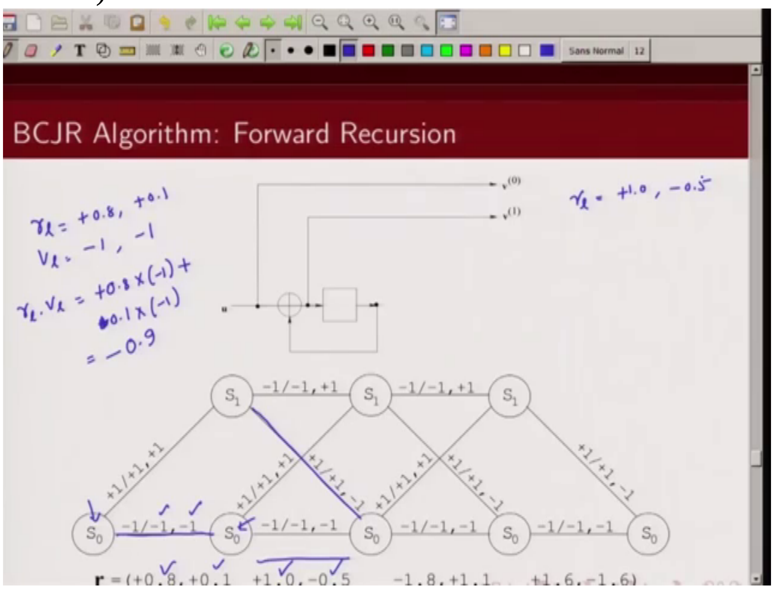
is time instance t equal to 1. so we were talking about this. And initial state is S 1; final state is S 0 so we are talking about this transition.

(Refer Slide Time 51:24)



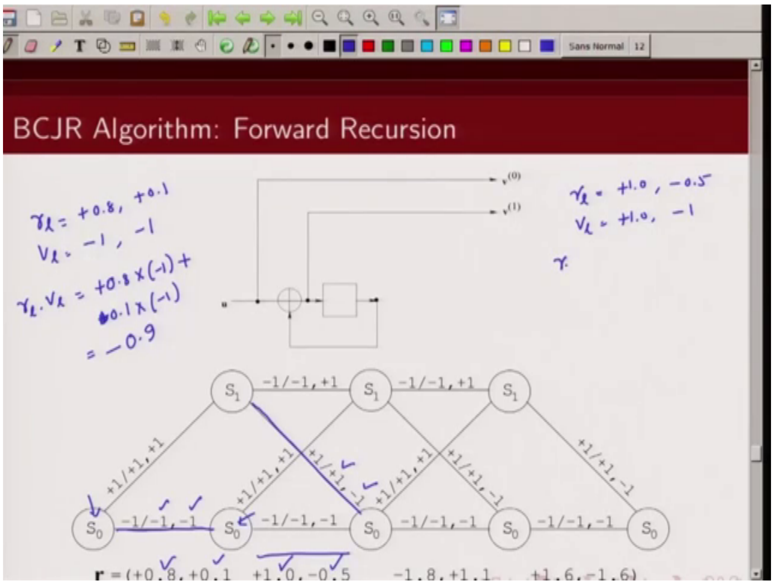
Now what is r l corresponding to this transition? It is plus 1 and minus point 5. So this is plus 1 and minus point 5. What is v l corresponding to

(Refer Slide Time 51:39)



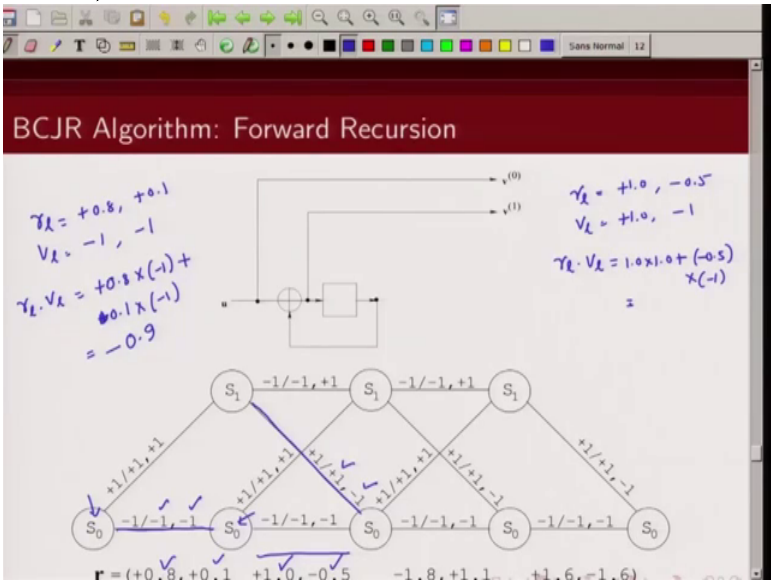
this transition? It is given by plus 1 and minus 1, so v l is plus 1 and minus 1; So then what would be r l

(Refer Slide Time 51:52)



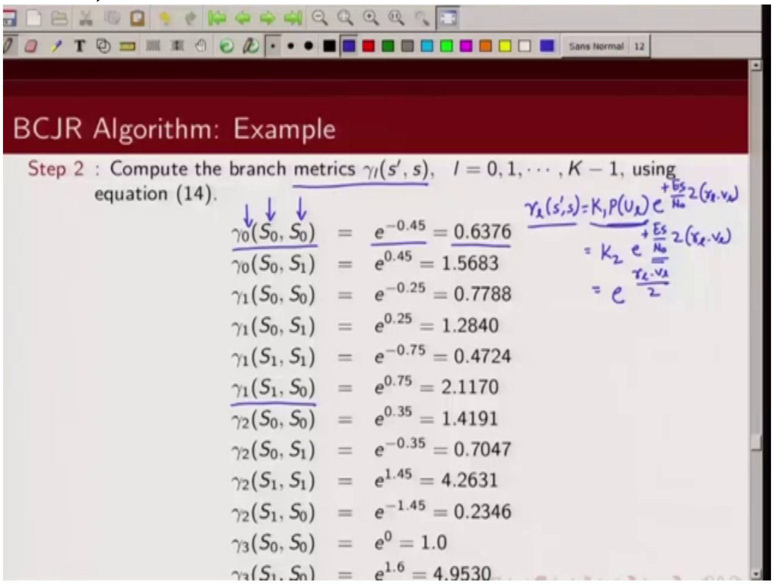
dot v l in this example? It is 1 into 1 plus minus point 5 into minus 1. So this will be

(Refer Slide Time 52:04)



1 point 5. If you plug that in here, 1 point 5 by

(Refer Slide Time 52:10)

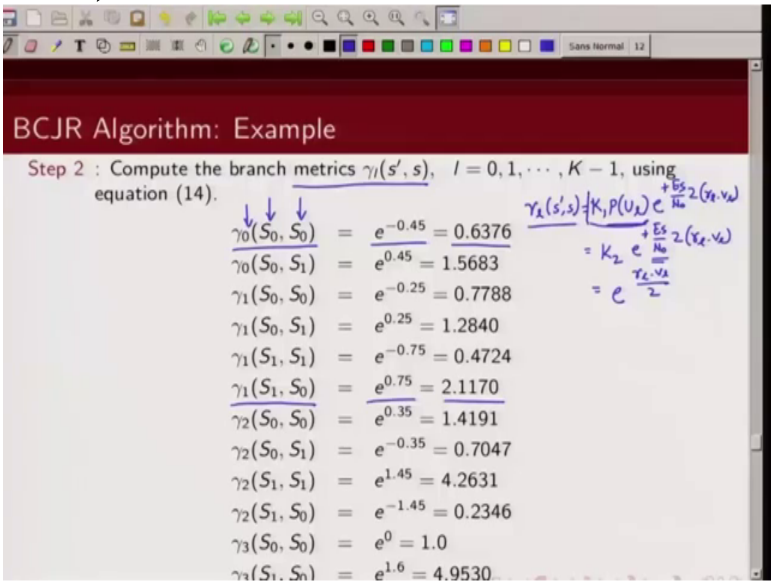


2, so this would be e raised to power point 7 5. And that's what it is, Ok. So I hope it's clear how we can compute the branch metric.

(Refer Slide Time 52:22)

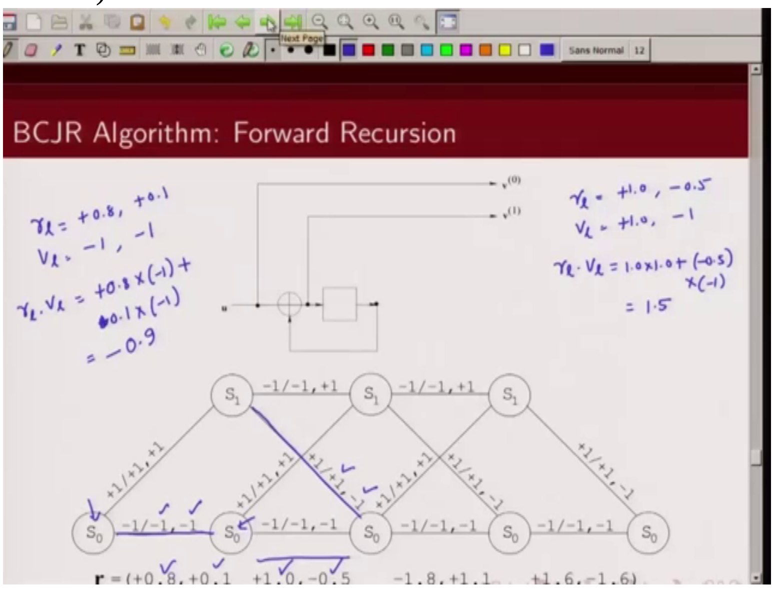
Please note we have ignored the common term

(Refer Slide Time 52:25)



which is common for computation of all of them. We are just computing the term which would be different based on what state transition that we are considering. So similarly we can compute gammas for other time instance t equal to 1, t equal to 2 and for all other valid state transitions. Now the next step is

(Refer Slide Time 52:49)

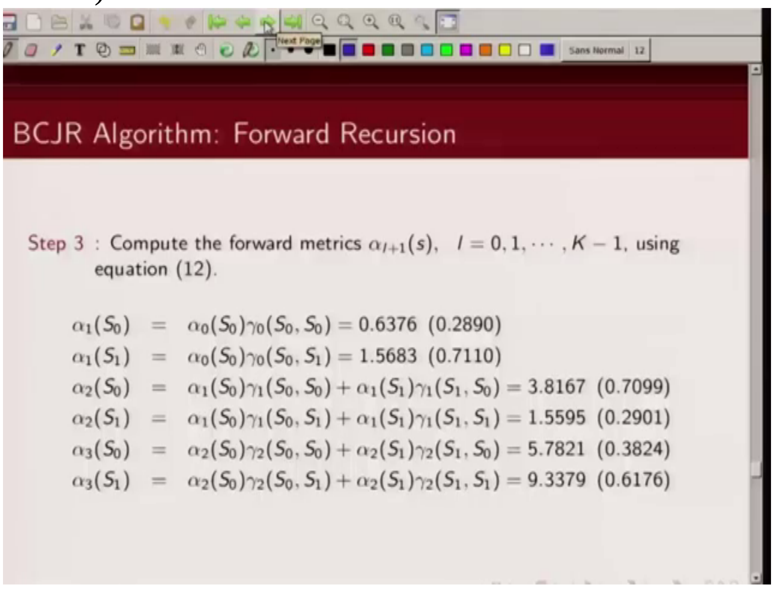


once we know our gamma we have already initialized our alphas and betas so the next

(Refer Slide Time 52:55)

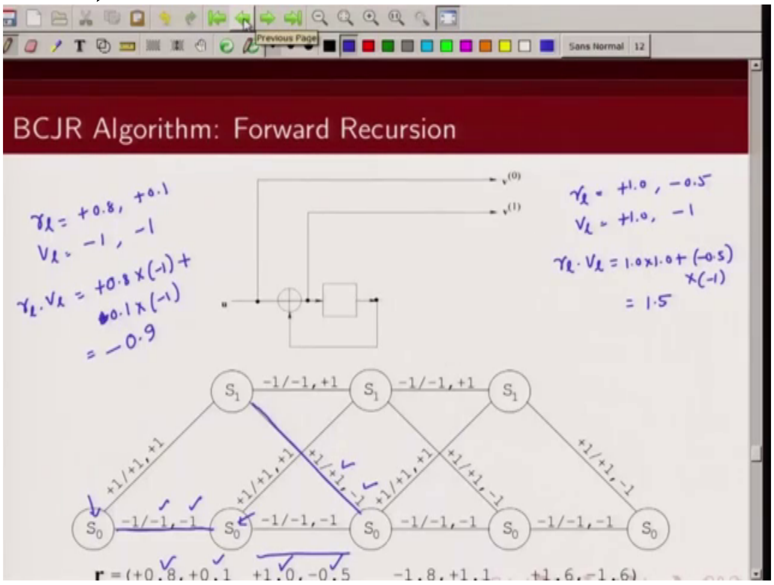
step would be to compute alphas and betas. And this is shown here. Now we have already illustrated how we can compute our alphas and betas. When we

(Refer Slide Time 53:11)



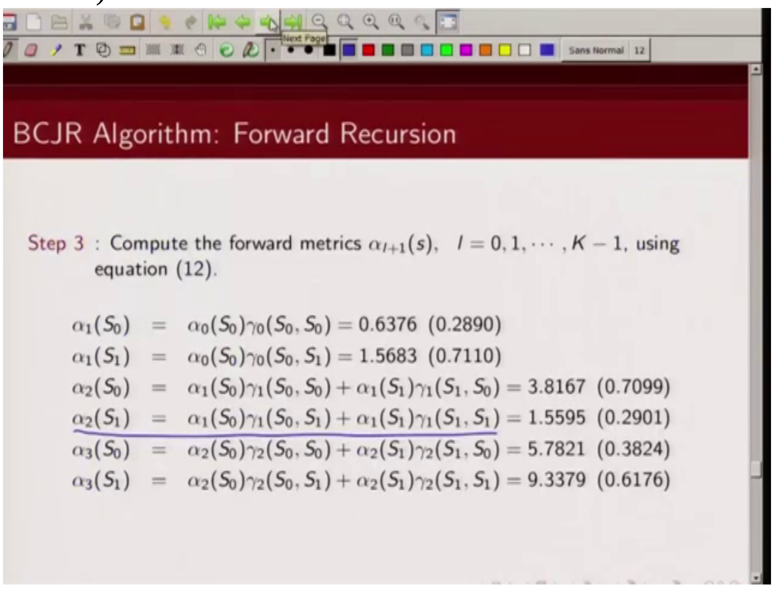
explained how to compute these forward recursion and backward recursion we can take another example. Let's just take this example. alpha 2 at state S 1, so alpha 2 will correspond to,

(Refer Slide Time 53:28)



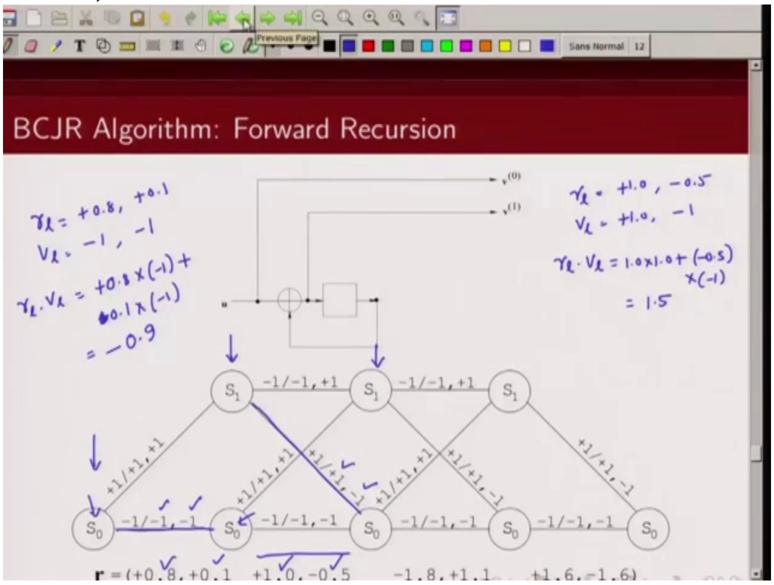
so this is alpha at 0, alpha at 1, so this is alpha at 2. So we are interested to calculate alpha 2 at

(Refer Slide Time 53:39)



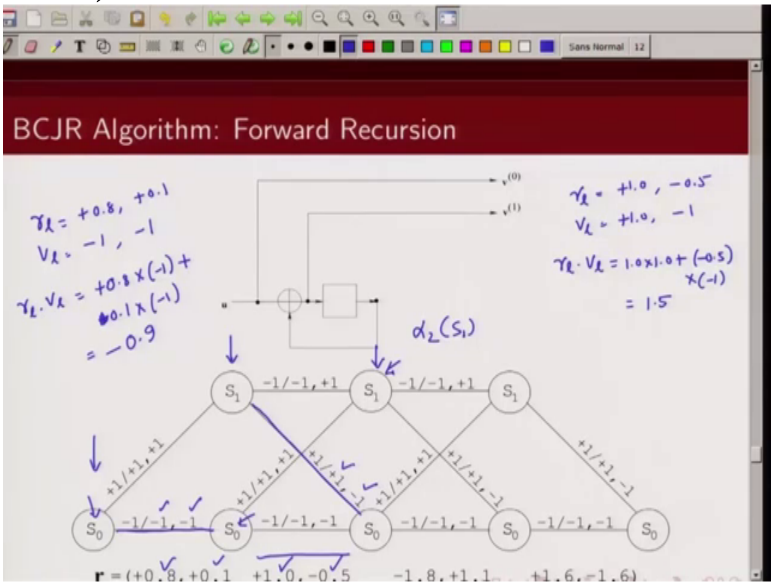
state S 1. So we are interested to calculate

(Refer Slide Time 53:42)



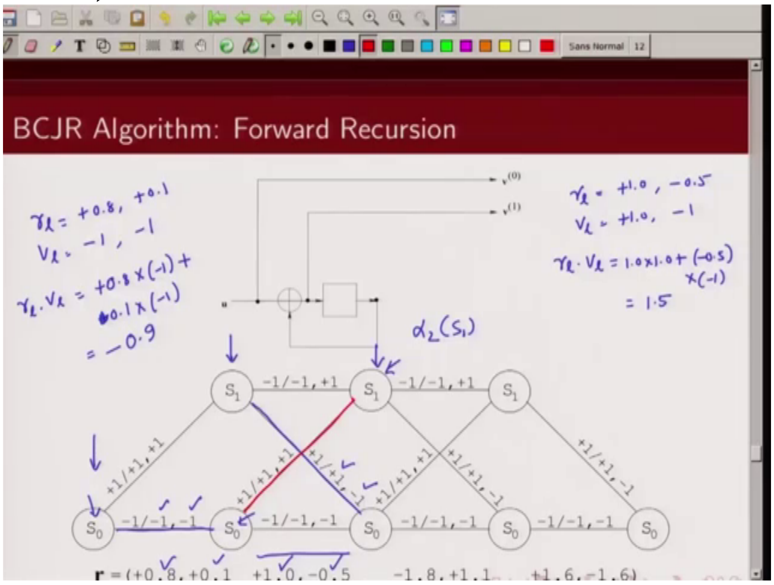
alpha 2 at state S 1. So we are interested to calculate alpha value here.

(Refer Slide Time 53:48)



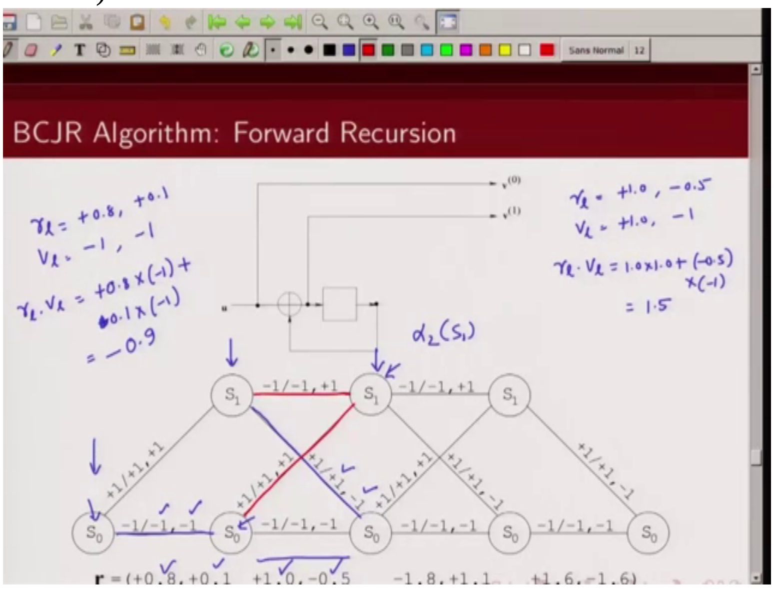
Now what are the transitions that are ending at this state? One is this,

(Refer Slide Time 53:57)



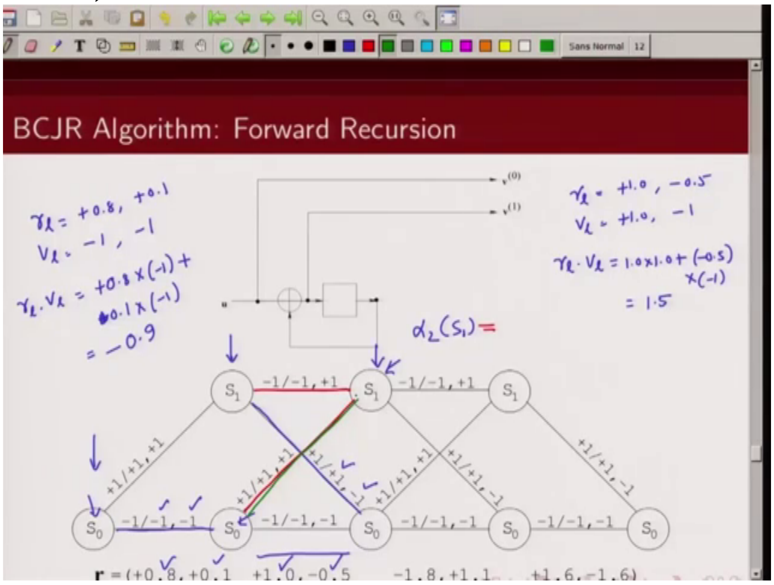
another is this.

(Refer Slide Time 53:59)



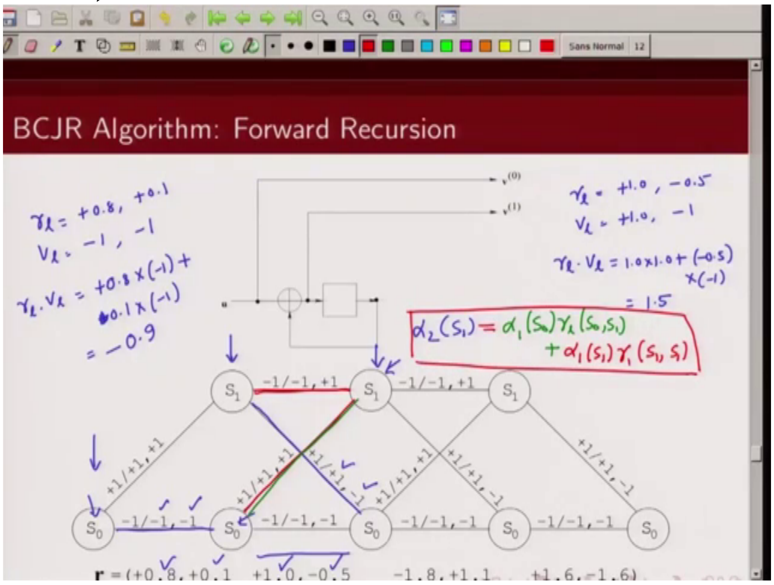
So there will be 2 terms in the alpha computation of this, one corresponding to this transition which is

(Refer Slide Time 54:11)



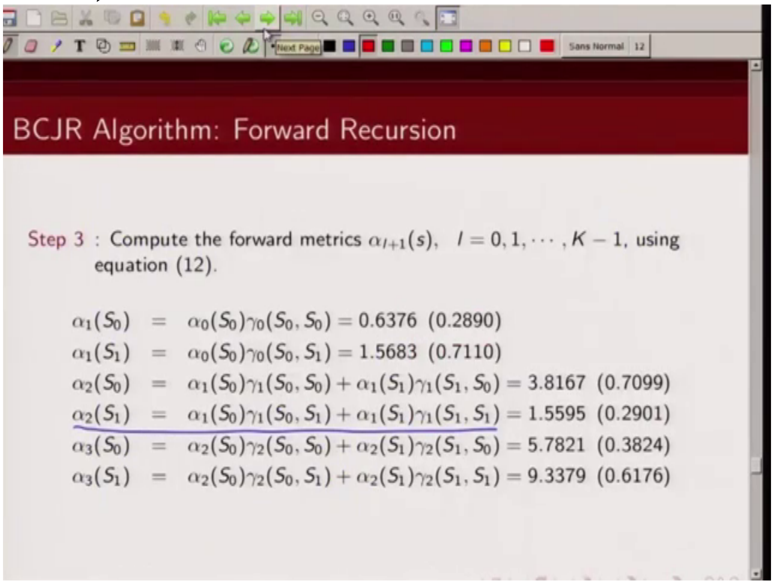
given by alpha at time 1, S 0 into gamma 1 S 0 S 1 plus there will be another term corresponding to this transition, this will be alpha 1 at state S 1 times gamma 1, initial state S 1 next state S 1. So this is the value of alpha 2 at state S 1

(Refer Slide Time 54:49)



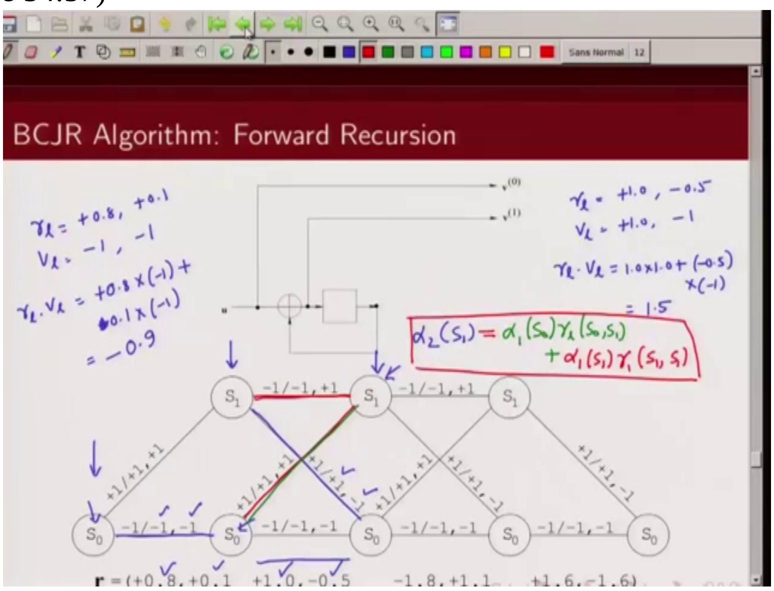
and that's what we have got here. You can check alpha 1

(Refer Slide Time 54:53)



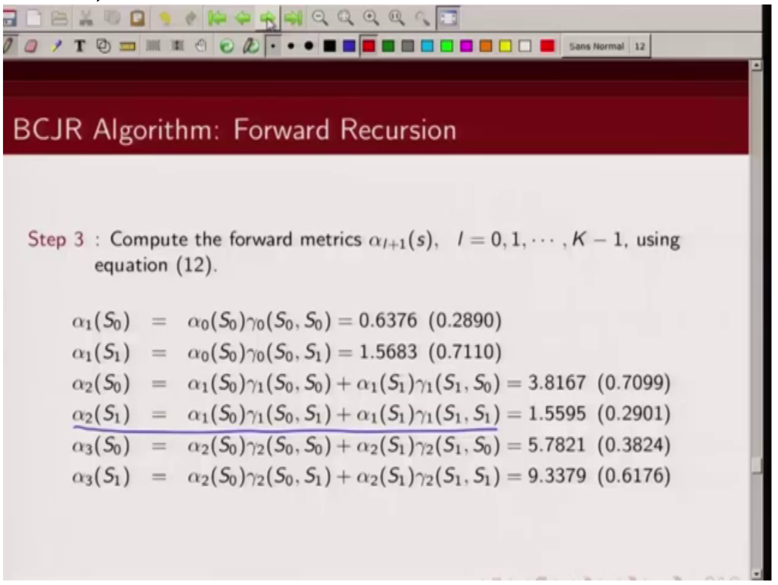
S 0, gamma S 0 S 1, gamma S 0 S 1

(Refer Slide Time 54:57)



and the next term is alpha S 1 gamma S 1 S 1,

(Refer Slide Time 55:02)

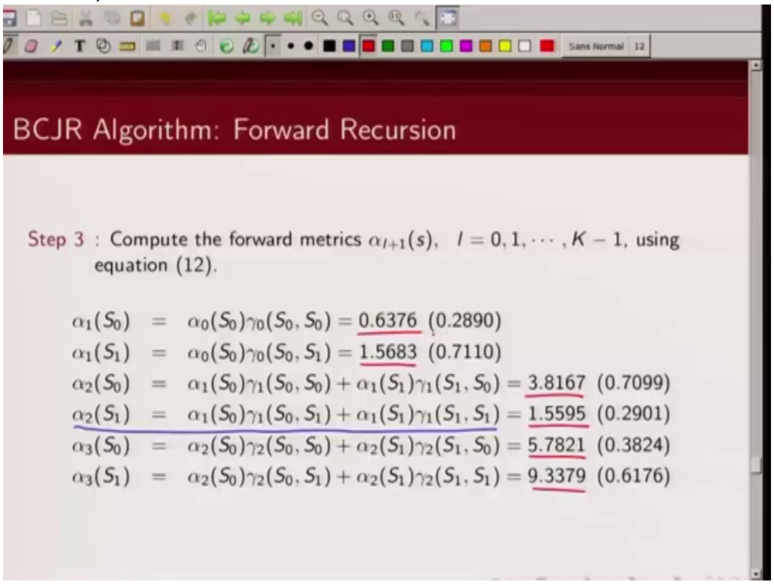


alpha S 1 gamma S 1 S 1. So like this we can compute the values of alphas. And these values that you see are basically the values of alpha computed this way. Now we can also normalize the values of alphas. Because alphas are, sum of alphas who are all

(Refer Slide Time 55:22)

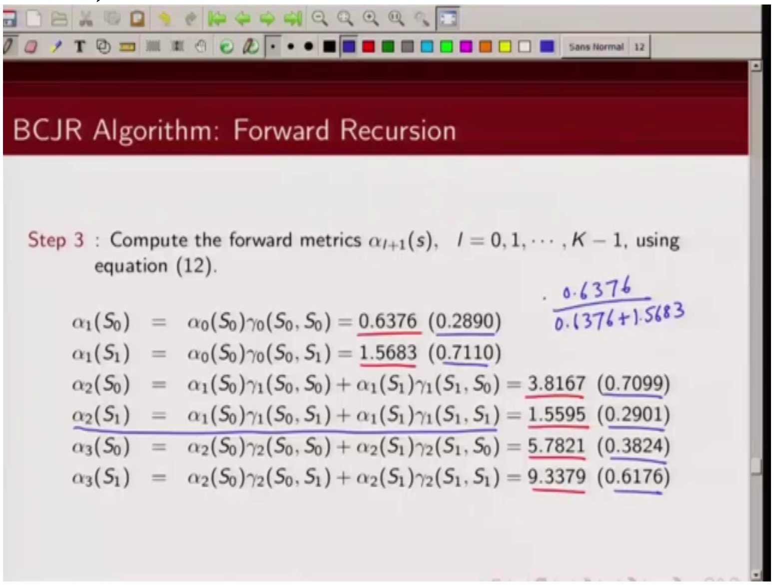
state should add up to 1. So in the bracket that you see here,

(Refer Slide Time 55:26)



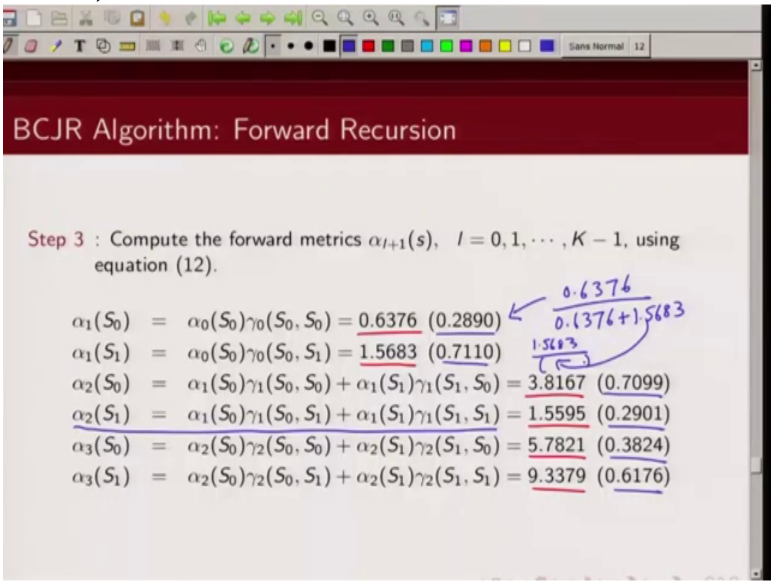
these are the normalized values of alpha. So how do I get this? So this is point 6 3 7 6 by point 6 3 7 6 plus 1 point 5 6 8 3. So this is this

(Refer Slide Time 55:45)



quantity. Similarly this is 1 point 5 6 8 3 divided by this one. So

(Refer Slide Time 55:57)

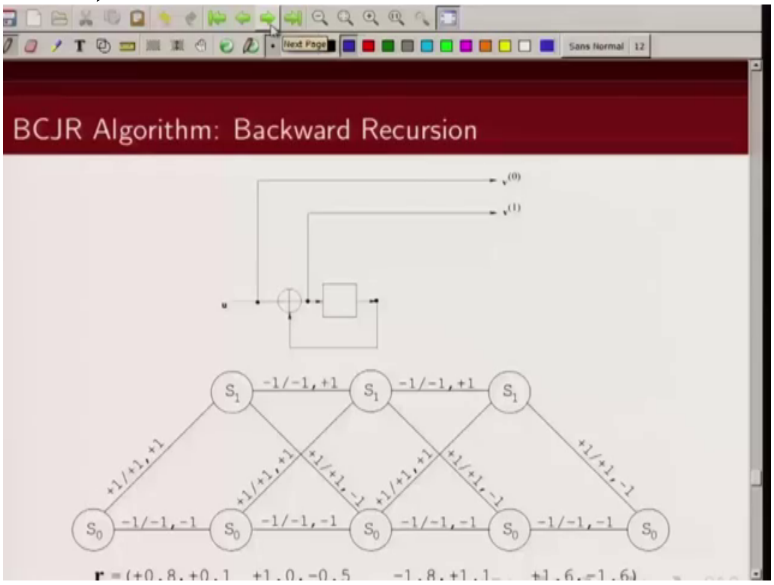


these are actual values of alpha and these are normalized values because these are, sum of probabilities should add up to zero so I can, I can take this value. So when I compute alphas or I can just work with these values or I can work with these values. It is just the scaled version so does not make a difference except for implementation purpose you may want to scale

(Refer Slide Time 56:22)

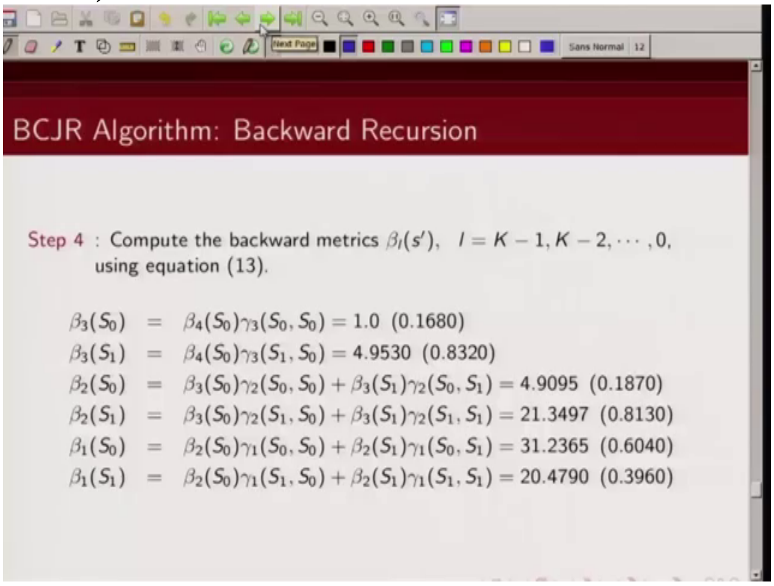
them, add them up to 1, so that the values do not blow up. Once we have computed

(Refer Slide Time 56:27)



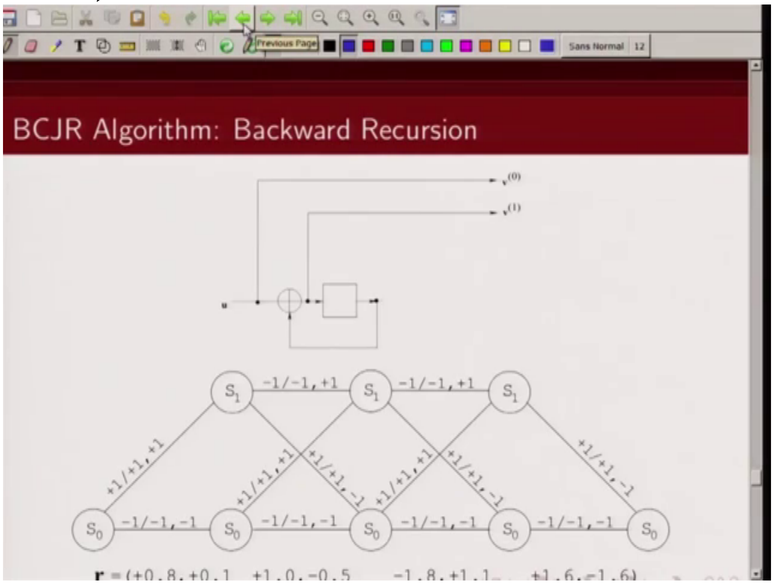
alphas, we can follow the same procedure to compute beta. So

(Refer Slide Time 56:33)



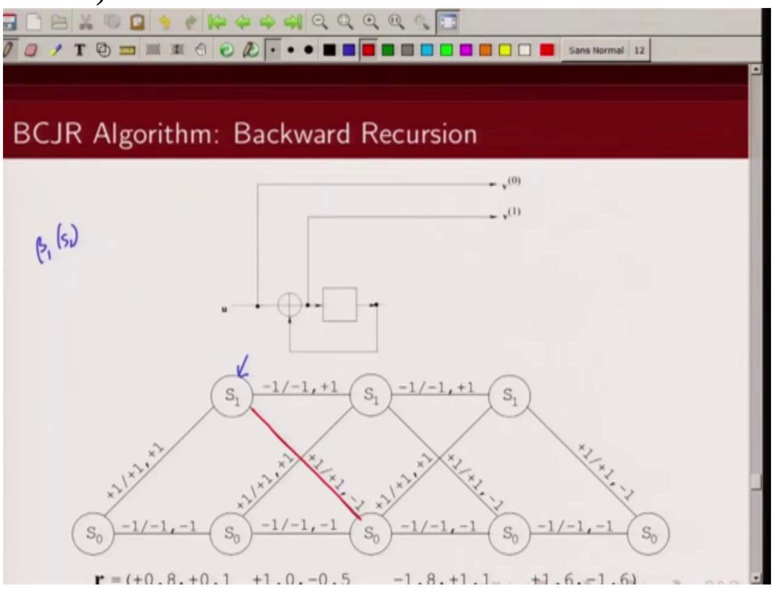
beta computation is given here. Again we can, as an example we can take one particular case. Let's just consider this. beta 1 at state S 1 so this is

(Refer Slide Time 56:45)



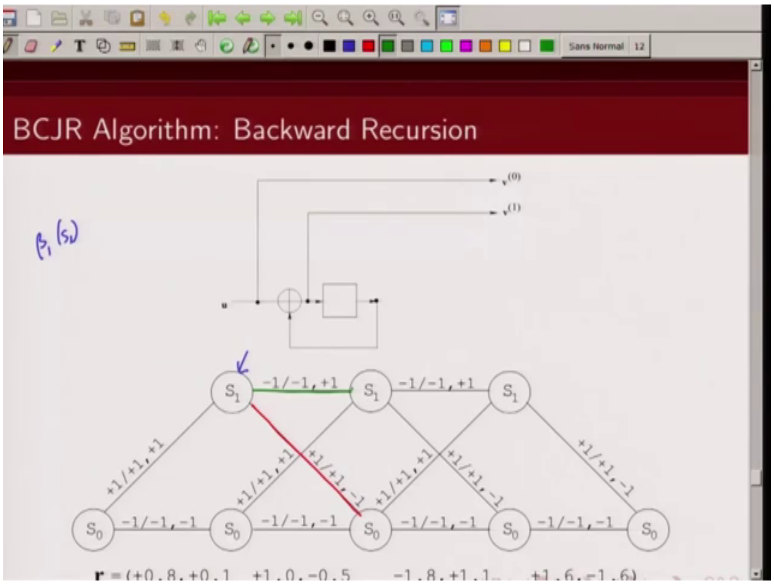
beta 0, beta 1 at state S 1 so we are interested in computing beta 1 at state S 1. Now what are the transitions that are ending in state S 1? One of them is this,

(Refer Slide Time 57:01)



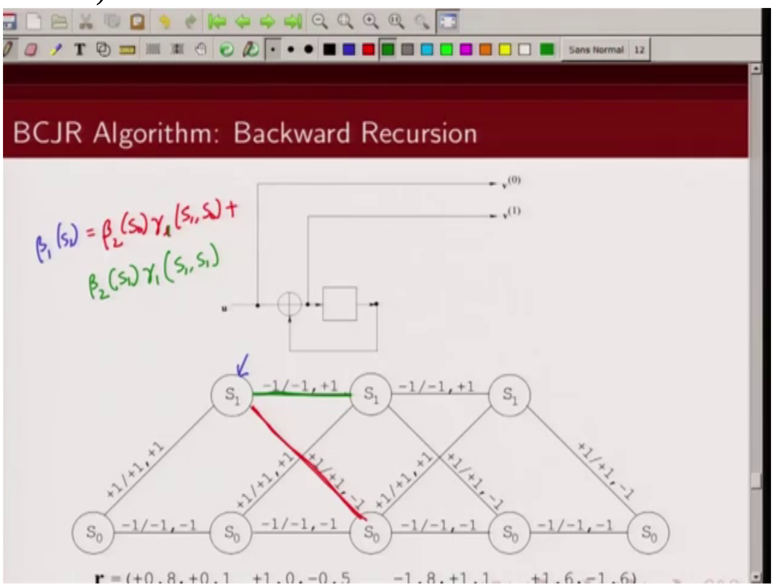
the another one is this. So

(Refer Slide Time 57:05)



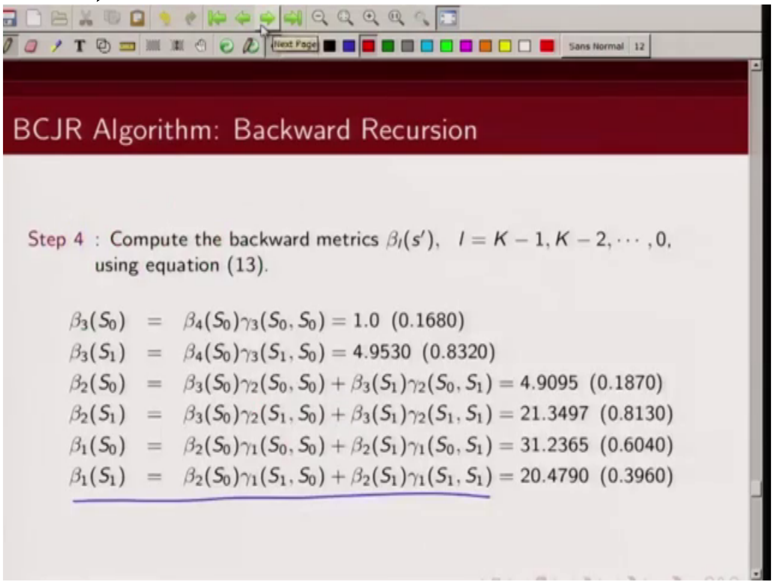
what is the contribution from this, this, this transition? This can be written as beta 2 S 0 times gamma 1 S 1 S 0 plus, and the contribution from this will be beta 2 S 1 times gamma of this, this is gamma 1, this is 1, gamma 1 of S 1, S 1. So this, the one in green is corresponding to this transition,

(Refer Slide Time 57:48)



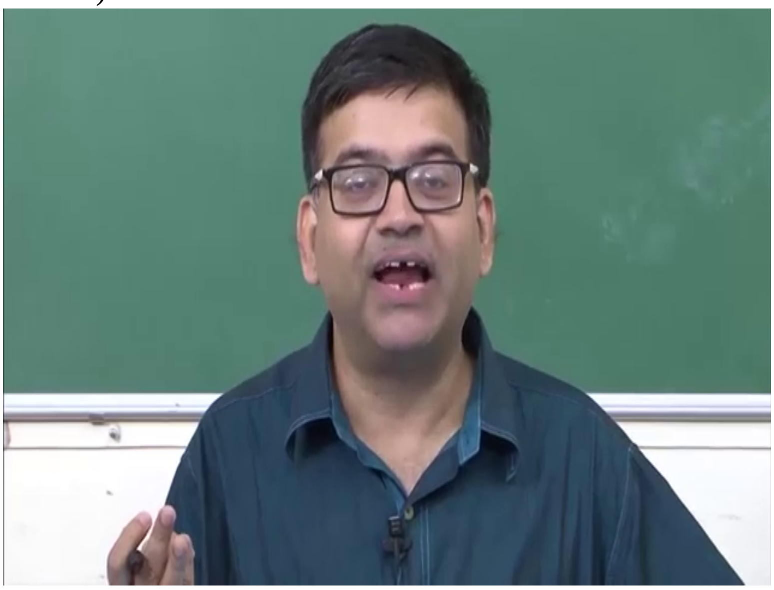
the one in red is due to this transition, Ok, this transition. So beta S 1 can be written as beta 2 S 0 gamma 1 S 1 S 0 plus beta 2 S 1 gamma 1 S 1 S 1. And that's what we have,

(Refer Slide Time 58:06)



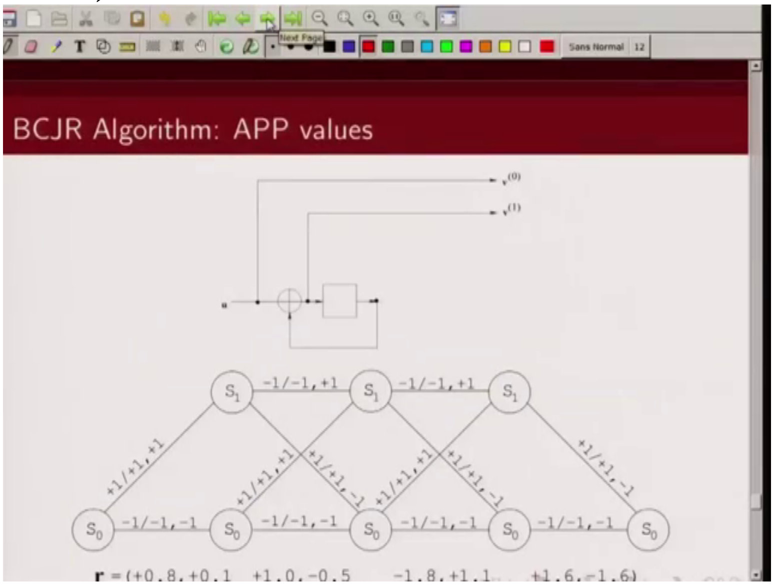
beta 2 S 0 gamma 1S 1 S 0, beta 2 S 1 gamma 1 S 1 S 1. We already know the values of gammas that we have computed earlier. And we know the values of beta at the end of block. What is this value? This is 1 because the encoder is terminated. And for all other state basically beta 4 S 1 is 0, so this is 1. We know these values so we can compute what is beta 3. Once we know the values of beta 3, we can compute the values of beta 2 because beta 3's values we will require here. Once we know the values of beta 2, we can compute the value of beta 1. They are required here. So like that basically we can recursively compute the values of betas. So now what do we have? We have the values of alphas, we have the values of betas and we have the branch metric

(Refer Slide Time 59:04)



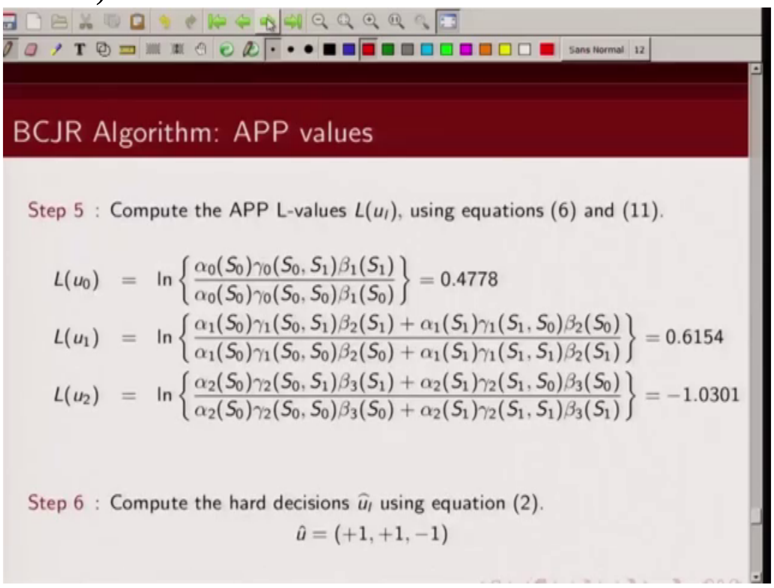
gammas. Next step we need to compute the A P P value. And what is the A P P value?

(Refer Slide Time 59:12)



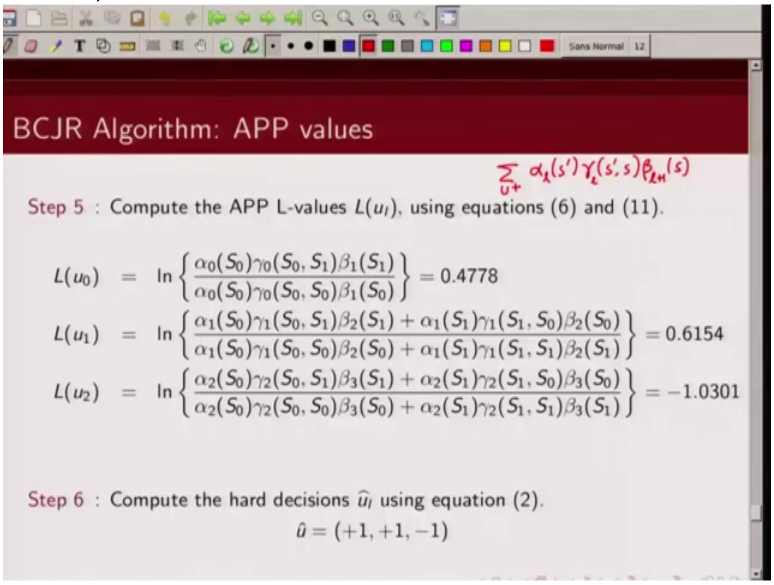
If you recall

(Refer Slide Time 59:14)



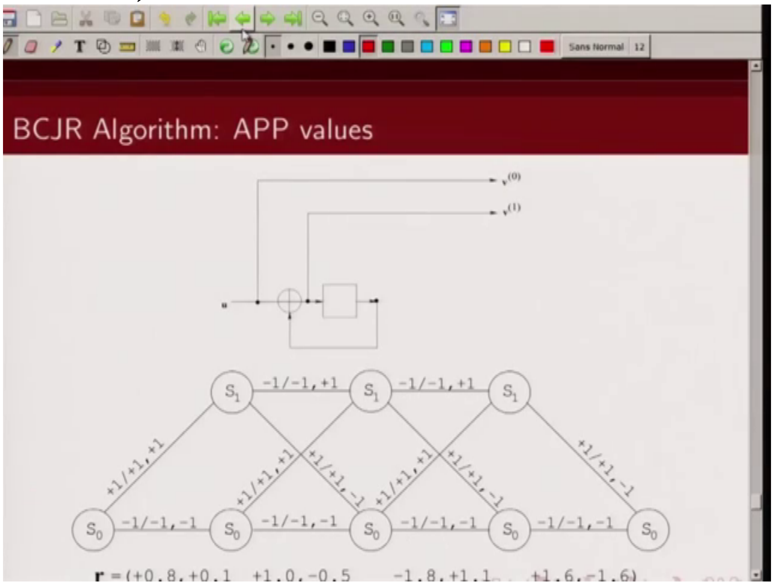
A P P value, it was product of three terms, alpha, gamma and betas. It's a product of this term. And what was the term in the numerator? We were summing over all those transitions which belongs to u being plus 1 and in the denominator

(Refer Slide Time 59:47)



we were summing over all those transitions belonging to u being minus 1. So let's look at how we can compute this. So let's look at first case.

(Refer Slide Time 01:00:12)

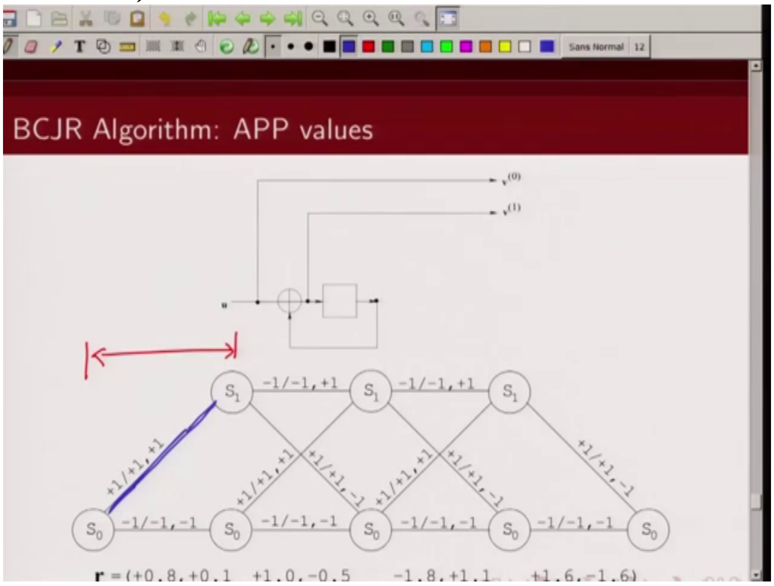


At time, so u 0 is the information sequence that we are trying to estimate for time, first time instance t equal to zero so this is we are looking at this Trellis section. Now which are the transitions corresponding to u l being

(Refer Slide Time 01:00:23)

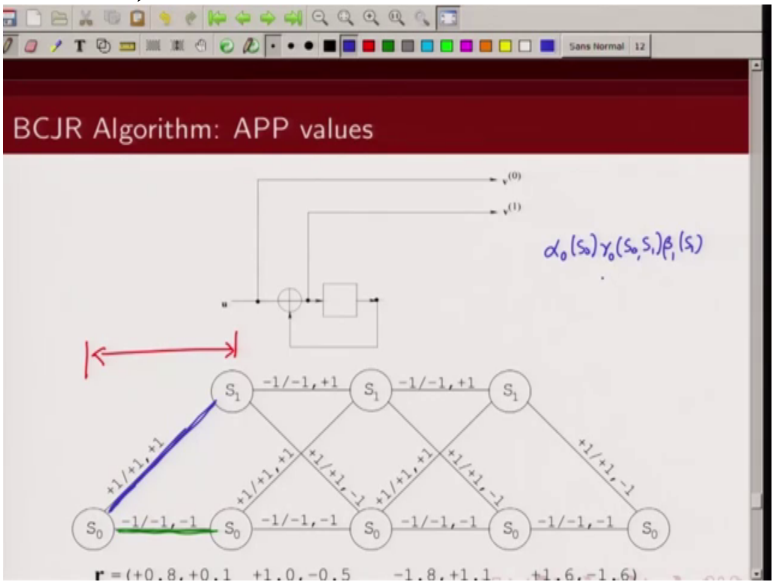
plus 1? So I am denoting by blue, the transitions which is corresponding to

(Refer Slide Time 01:00:31)



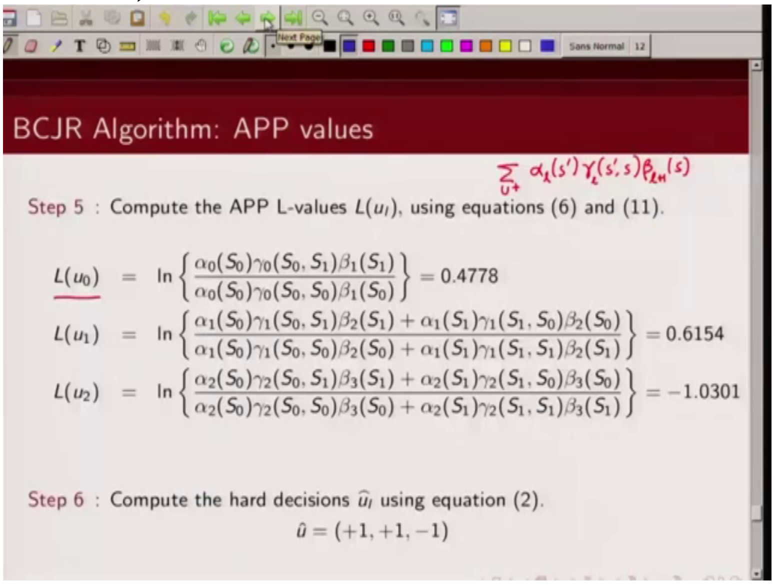
u l being plus 1 and I am denoting by green the transition corresponding to u l being minus 1, Ok. Then in the numerator then, I will have one term corresponding to this transition and what would be that term? It would be alpha 0 S 0 and gamma corresponding

(Refer Slide Time 01:01:14)



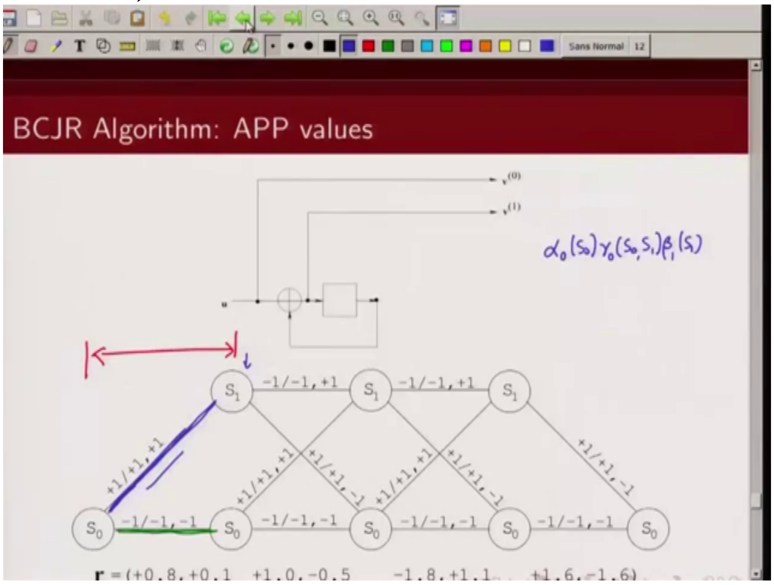
to this transition which is gamma 0 S 0 S 1 times beta 1 S 1. So I have alpha 0 S 0, alpha at this state, gamma corresponding to this transition, which is gamma 0 S 0 S 1, and beta corresponding to this state, if we go back what I have,

(Refer Slide Time 01:01:26)



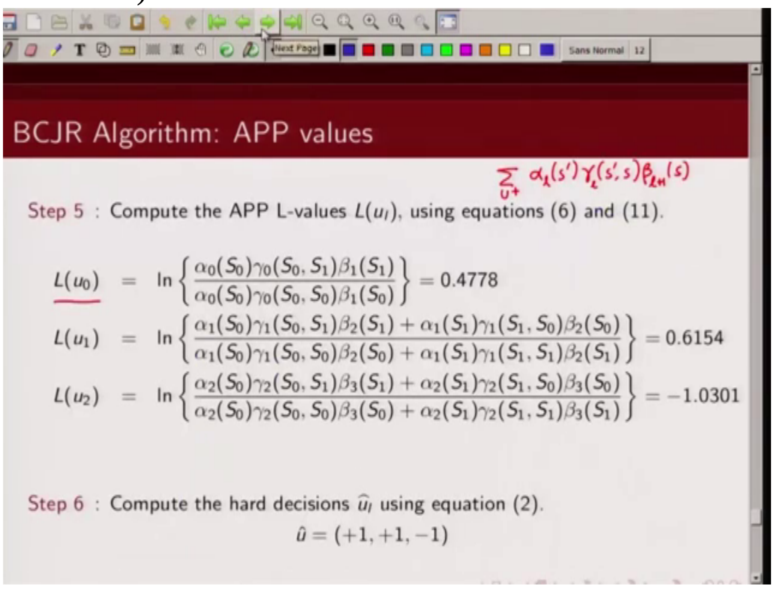
alpha 0 S 0, gamma 0 S 0 S 1 and beta 1 S 1. At this time instance

(Refer Slide Time 01:01:35)



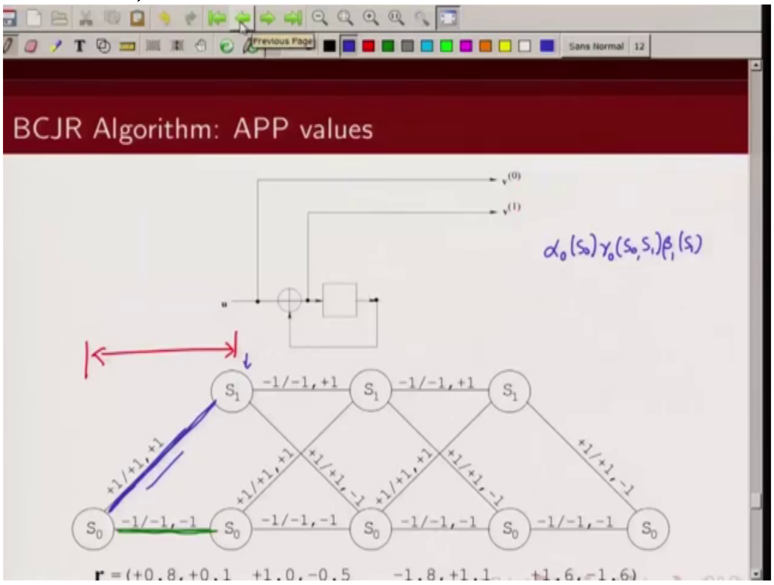
is there any other transition corresponding to u l plus 1? No. It is only one transition corresponding to u l plus 1. So we will now look at

(Refer Slide Time 01:01:47)



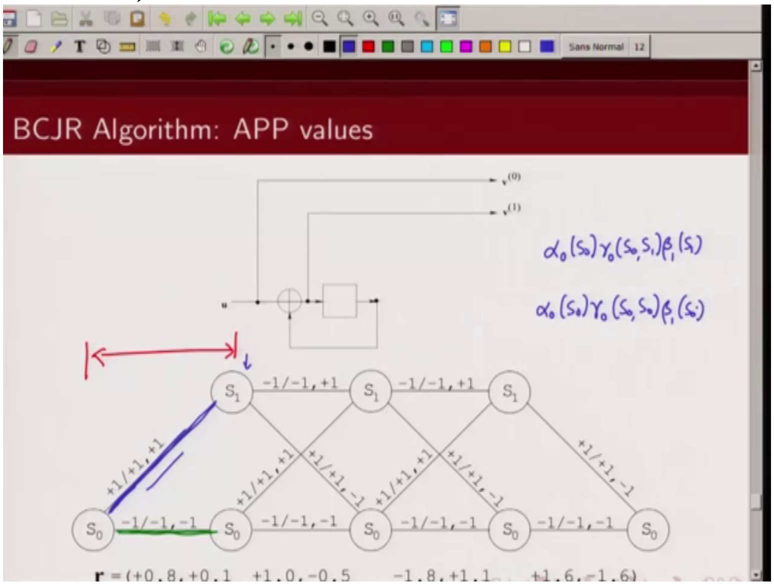
the denominator term. So we have to look for all those transitions corresponding to u being minus 1 and there is only one such

(Refer Slide Time 01:01:56)



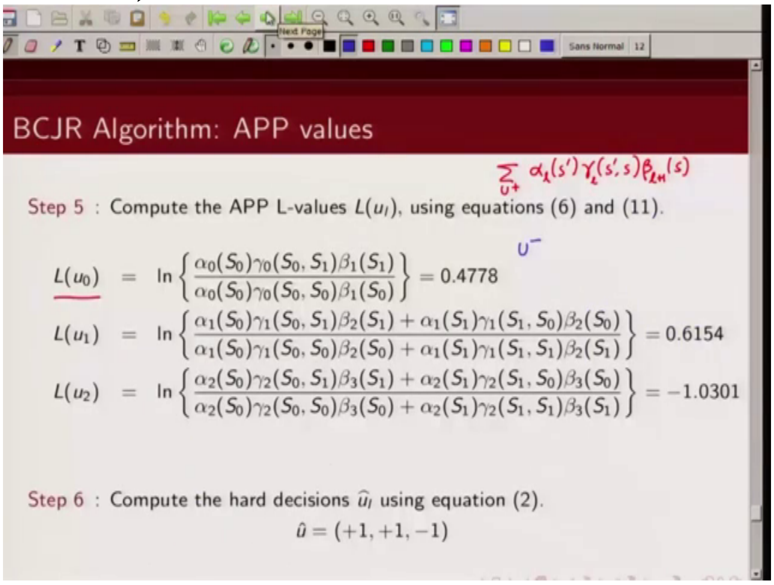
transition so the denominator term would be alpha 0 S 0 gamma 0 S 0 S 0 and beta 1 S 0. So this

(Refer Slide Time 01:02:12)



is what we have,

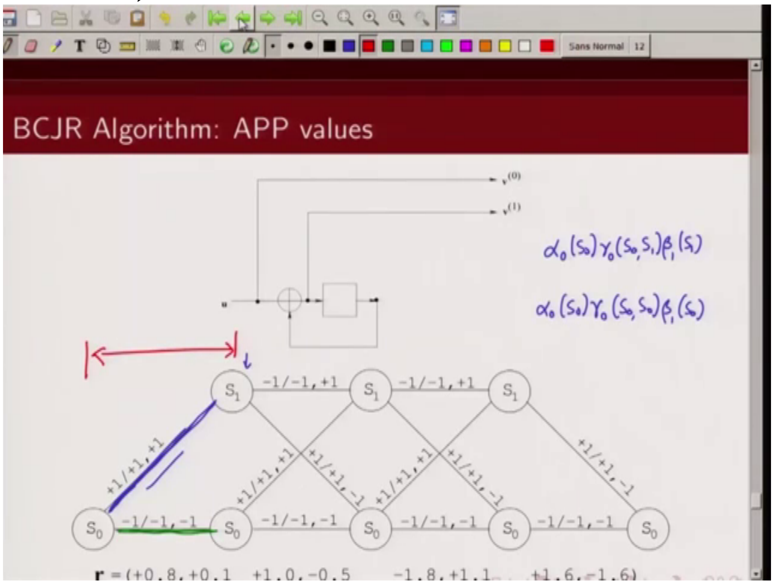
(Refer Slide Time 01:02:13)



alpha 0 S 0 gamma 0 S 0 S 0 and beta 1 S 0. So we can then calculate the what's the a posteriori L value.

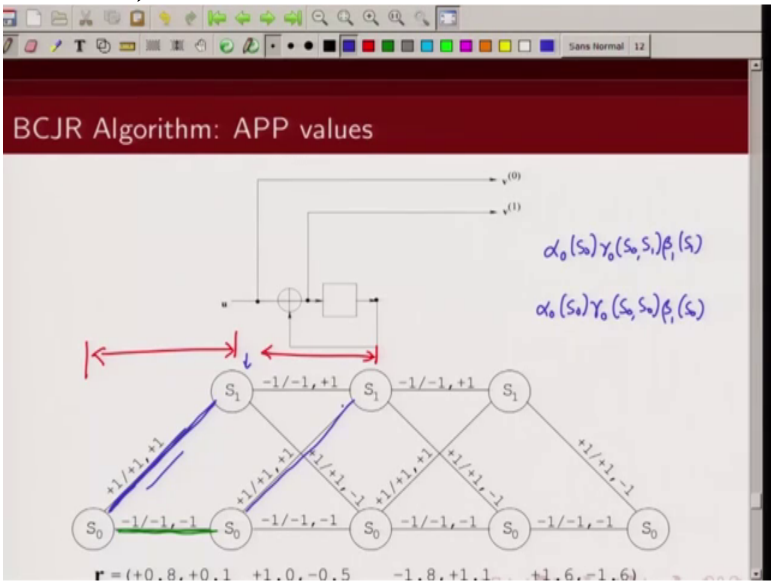
Now let's take another example. Let's take for the second time instance. So the second time instance we are interested in estimating what was our information sequence, information bit. So

(Refer Slide Time 01:02:45)



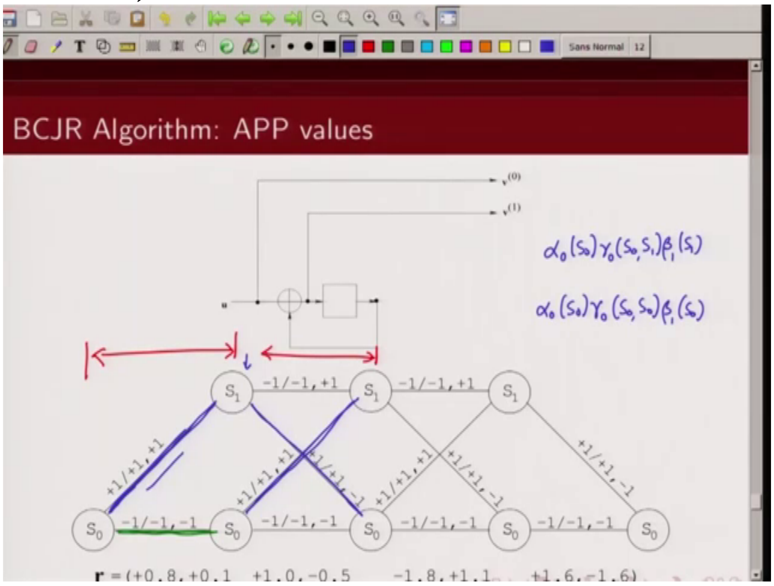
we are now looking at this time instance, this time instance, Ok. Now what are the transitions corresponding to u l, information sequence being plus 1? One of them is this. You can see

(Refer Slide Time 01:03:04)



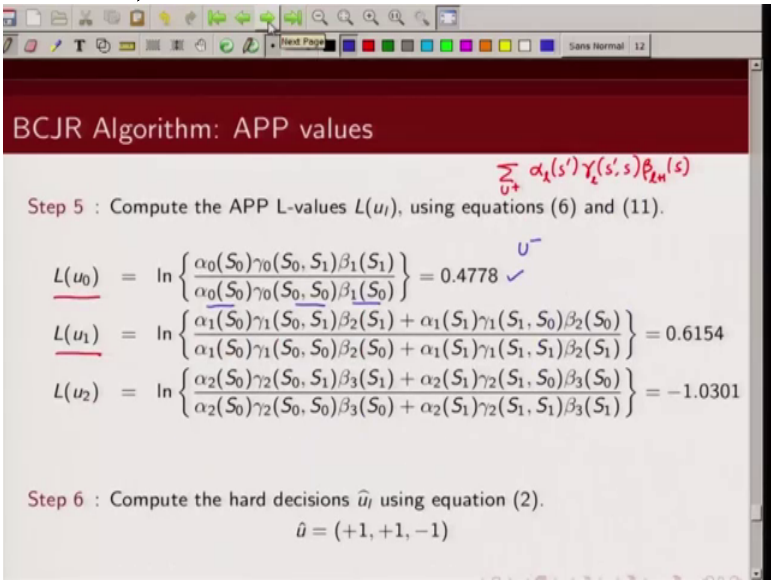
the information sequence is plus 1, that's when you go from S 0 to S 1. And another transition is this one. So

(Refer Slide Time 01:03:15)



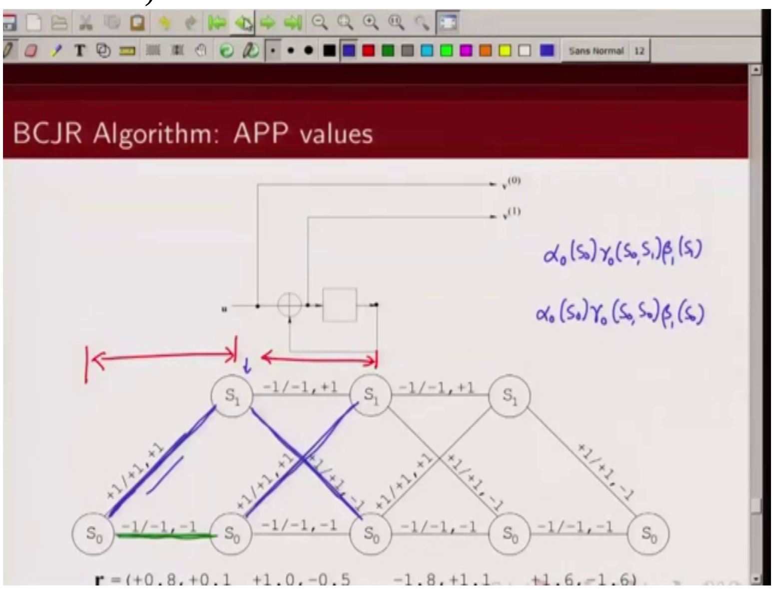
these are the two transitions corresponding to u l being plus 1. So in the numerator you will have 2 terms. One corresponding to alpha 1 S 0 gamma 0 S 0 S 1 times beta 2 S 1 and another term corresponding to this transition which is alpha 1 S 1 times gamma 1 S 1 S 0 multiplied by beta 2 S 0 and that's what you see here.

(Refer Slide Time 01:03:52)



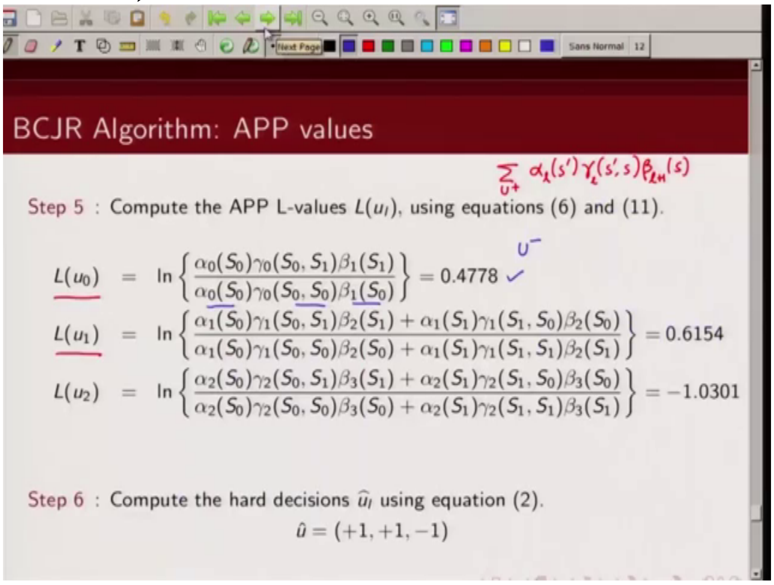
There are 2 terms, one is alpha 1 S 0 gamma 1 S 0 S 1 beta 2 S 1 this corresponds to this transition and the next term that you see here, There are 2 terms, one is alpha 1 S 0 gamma 1 S 0 S 1 beta 2 S 1 this corresponds to

(Refer Slide Time 01:04:01)



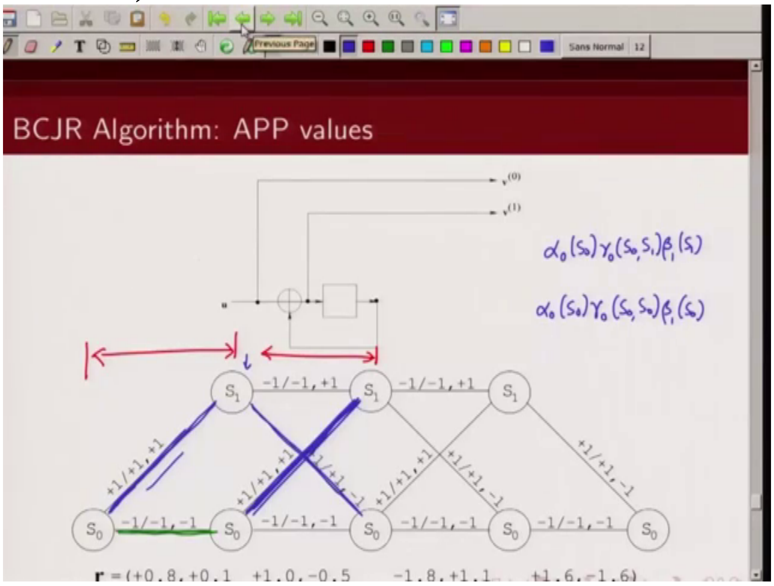
this transition and the next term that you see

(Refer Slide Time 01:04:07)



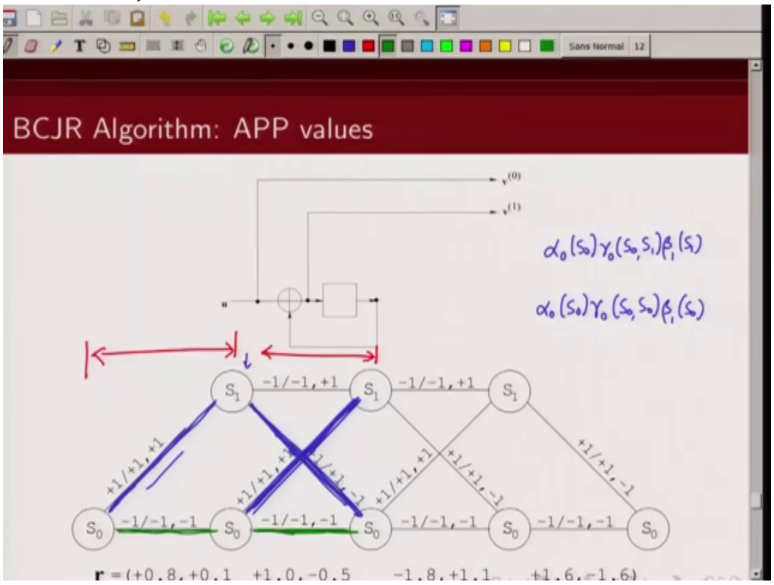
here, this one corresponds to

(Refer Slide Time 01:04:11)



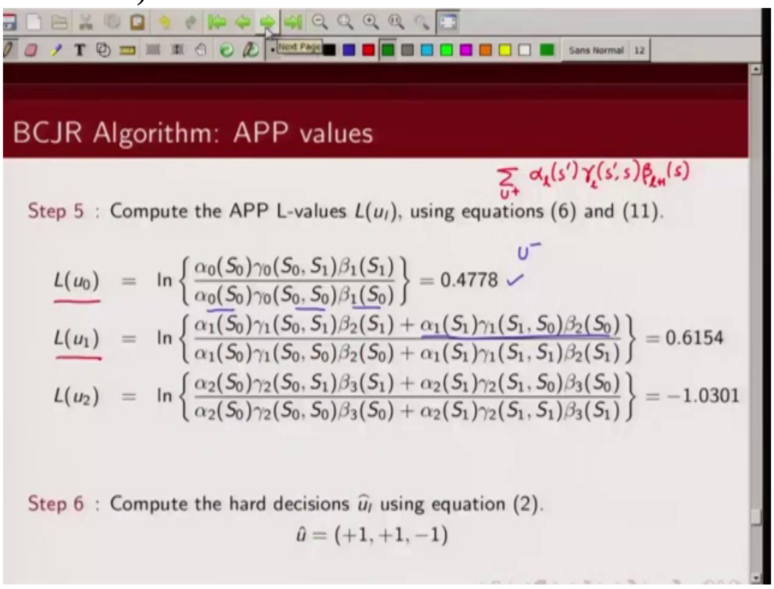
this transition, Ok. Now similarly in the denominator you need to look at what are the valid transitions corresponding to u l minus 1 and what are those? One of them is this

(Refer Slide Time 01:04:29)



and the second one is this. So now in the numerator, denominator also you will have two terms, one corresponding to this transition, other corresponding to this transition, this transition will give you alpha 1 S 0 gamma 1 S 0 S 0 times beta 2 S 0 plus alpha 1 S 1 gamma 1 S 1 S 1 times beta 2 S 1. And that's what you have here.

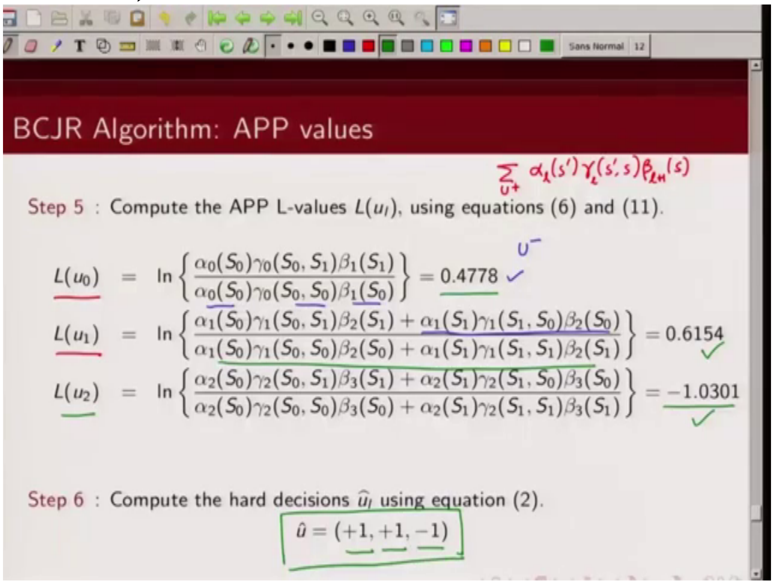
(Refer Slide Time 01:05:04)



So likewise we compute log likelihood ratios, A P P values for all the three information bits. Now what is the final step? Once we have computed the log likelihood ratio we will see whether these log likelihood ratios are greater than 0 or less than 0. If they are greater than equal to 0, we decide in favor of u l being plus 1, otherwise we decide in favor of u l being minus 1. So this is point 4 7 7 8, which is greater than zero so we decide in favor of plus 1.

This is greater than zero so we decide in favor of plus 1 and this one is less than zero so we decide in favor of minus 1. So then the final decoded bits are plus 1,

(Refer Slide Time 01:06:00)



plus 1 and minus 1. So with this I will conclude this lecture, thank you.

(Refer Slide Time 01:06:08)