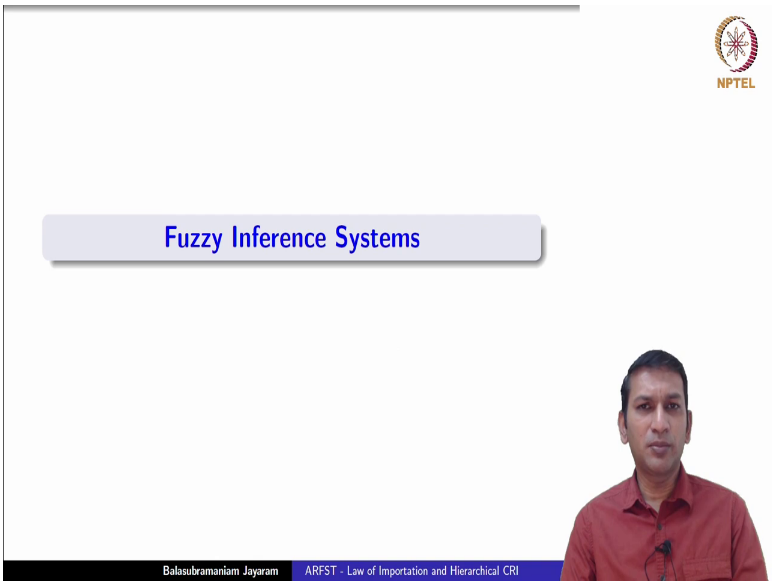
Approximate Reasoning using Fuzzy Set Theory

Prof. Balasubramaniam Jayaram
Department of Mathematics
Indian Institute of Technology, Hyderabad

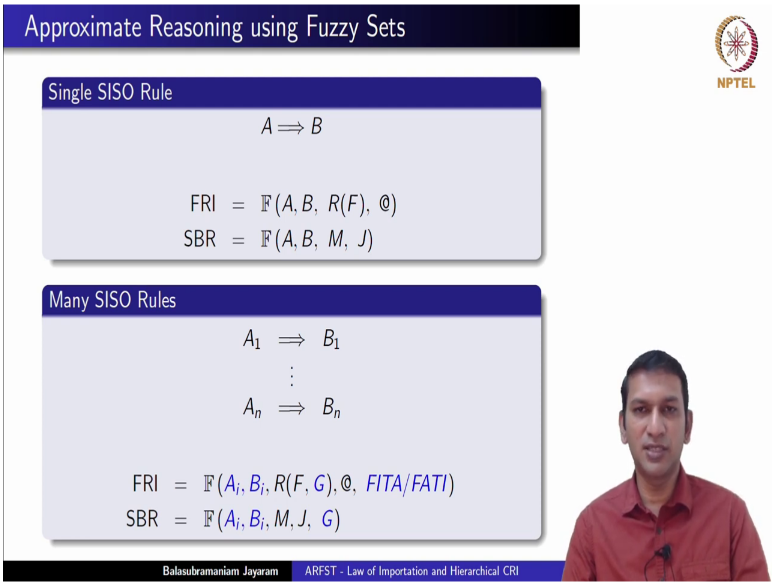
Lecture - 57
Law of Importation and Hierarchical CRI

Hello and welcome to the next of the lectures in this 12 week of the course titled Approximate Reasoning using Fuzzy Set Theory, a course offered over the NPTEL platform. In this lecture, we will look at one particular functional equation that of the Law of Importation and see how it enables us to obtain a computationally efficient procedure in getting an output from the compositional rule of inference.

(Refer Slide Time: 01:01)



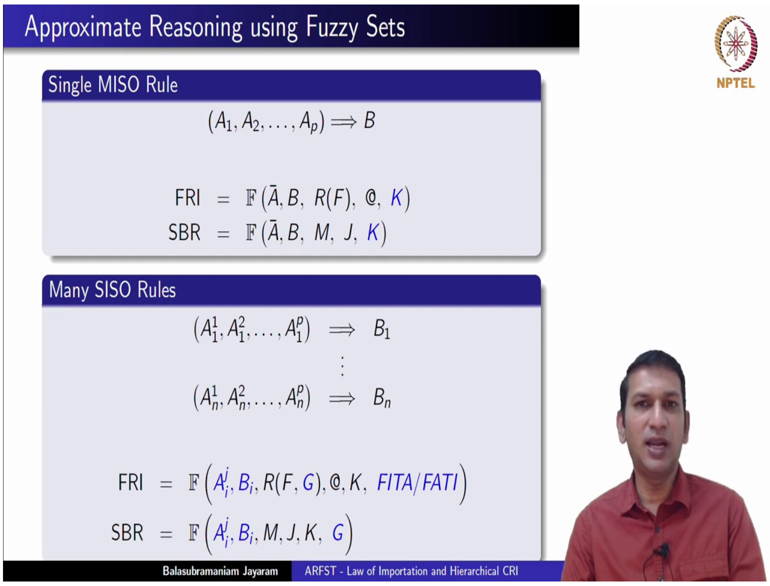
(Refer Slide Time: 01:05)



Let us look a begin by looking at fuzzy inference systems themselves. In the case of a single SISO rule, we have this A implies B; this is the rule that is given to us. If you are using an FRI all we need is a function F to relate the antecedent to the consequents and obtain a relation and a composition operator. In the case of an SBR scheme we need a matching function and also a modification function.

If you have multiple SISO rules, then, we also need an aggregation function G in the case of an FRI and also either the FITA or the FATI inference strategy. Whereas, in the case of an SBR, we will only need the aggregation function G, as extra compared to having a single SISO.

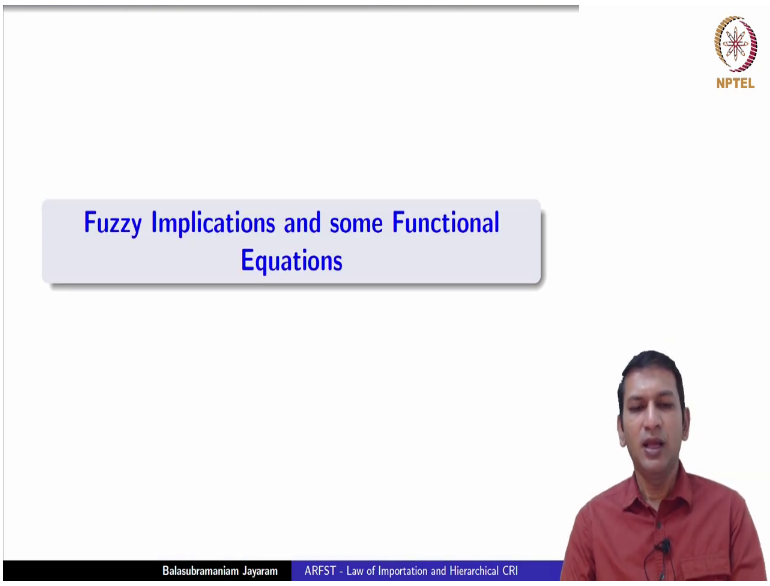
(Refer Slide Time: 01:53)



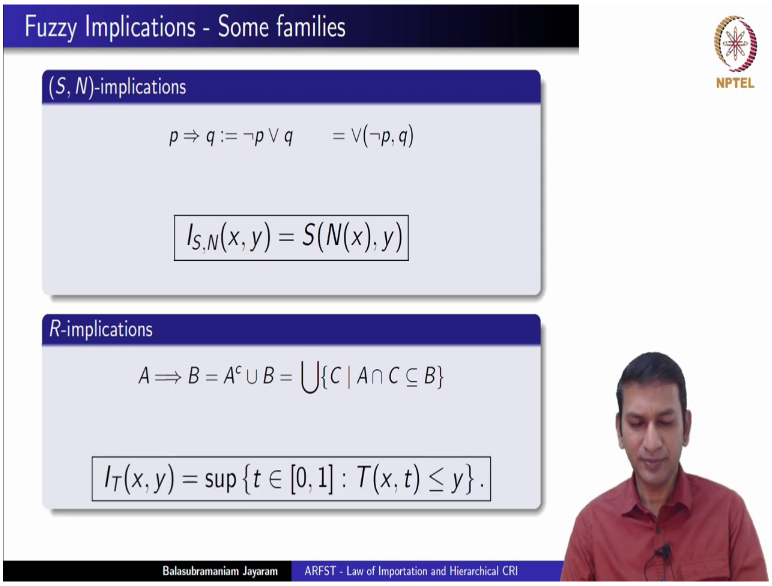
Now, if you have a single MISO rule; that means, rule where you have p dimensional inputs p a single rule where, we have p antecedents relating to a single consequents. In the case of an FRI, we have this A bar indicating a multi-dimensional input, we only need a function F to relate the antecedent multi-dimensional antecedent to the consequents and a composition operator.

Of course, we need an antecedent combiner K to combine all of them to get a single relation. In the case of an SBR again, we have this matching and modification function of course, we also need the antecedent combiner. In the case of multiple MISO rules, not only do we need the antecedent combiner we also need the FITA, FATI inference strategy and the aggregation function G. And in the case of SBR, we need the aggregation function G and also the antecedent combiner k.

(Refer Slide Time: 02:52)

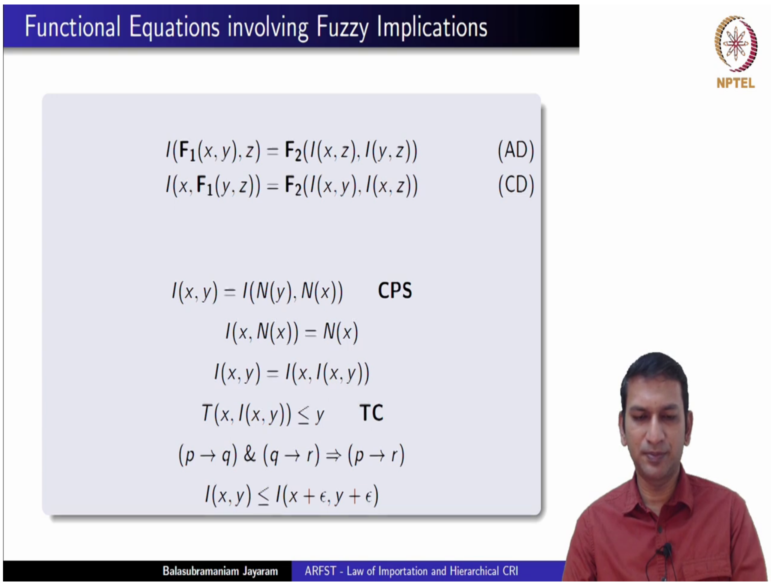


(Refer Slide Time: 02:57)



Let us look at some of the functional equations involving fuzzy implications. Let us look at the basic families of implications. The S, N family, which is S, N-implication we know about the R-implications which we have discussed at length almost for more than half of this course.

(Refer Slide Time: 03:15)



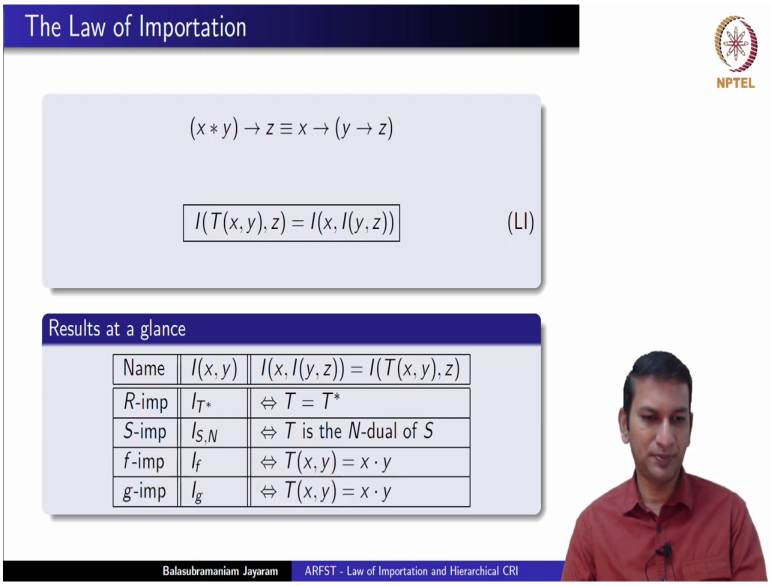
If you look at the functional equations, we have seen a couple of them in the previous lecture; which is the antecedent and consequents distributivity. But these are only two such of course, law of importation is one. But there are also many other functional equations involving fuzzy implications.

For instance, x implies y is equivalent to negation y implying negation x. This is the contrapositive symmetry. So, if we could also think of this as a functional equation. This is another functional equation I of x comma N x is equal to N x and this is called the iterative functional equation I of x, y is equal to I of x I of I x, y. This is the T-conditionality that we have seen this is called the hypothetical syllogism.

We have kept it in the almost logical format; because we could put I for the arrow and T for the ampersand here, but this is kept in this form so that we could see what exactly it is doing. It is essentially chaining inferences, p in implies q and q implies r implies p implies r. So, in some sense it is a transitivity property. It is just what is called as a hypothetical syllogisming usual logic setting.

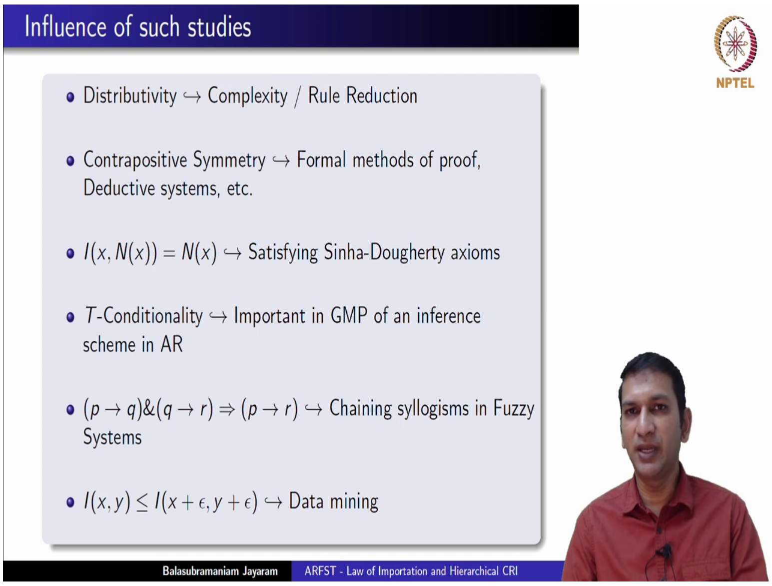
There is also this inequality I of x, y less than or equal to x plus epsilon y plus epsilon for admissible epsilons, this is called the specialty property. So, we have these many different functional equations involving fuzzy implications and often another binary operation or unary operation.

(Refer Slide Time: 04:48)



Of course, we also have the law of importation, which we have seen in this. And, we have seen in the previous lecture that for the major class of implications, we see that all of them do satisfy law of importation.

(Refer Slide Time: 05:07)



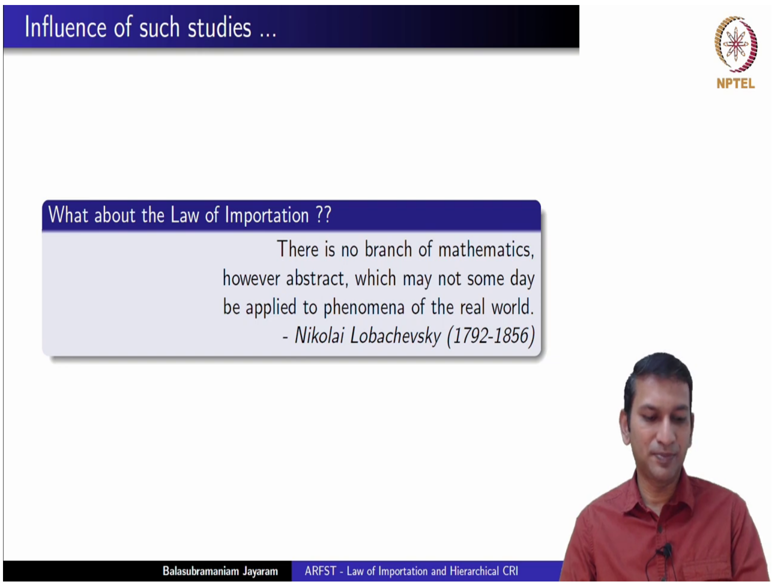
And we have in fact, characterized the T-norm with respect to which they will satisfy the law of importation. Now, why do we need to study such functional equations? Well, in the previous lecture we have seen, it allows us to generalize many settings especially the setting of inference mechanism itself by allowing the implications to come from families outside of the R-implication.

But that is not the only utility of studying such functional equations. For instance, it is well known that distributivity equations, if a implication has distributivity property; then in the inference scheme that such an implication is being used it can help in complexity or rule reduction. In the case of contrapositive symmetry as we all know, it is very important in deductive systems and formal methods of group.

Where, you want to show some result instead of showing A implies B you end up showing negation B implies negation A, the contrapositivity of the original statement. This is another functional equation which ensures that the Sinha-Dougherty axioms of subsethood measure hood; subsethood measure which are used in image processing will be valid.

So, an implication satisfying this functional equation is useful there. Of course, we know about T-conditionality, how it is important in generalized modest components of an inference scheme, especially when we discussed about a single SISO rule. This hypothetical syllogism is again useful and chaining syllogisms is fuzzy systems. And, this inequality which is called the special inequality such implications are called special fuzzy implications. They are shown to be useful in some data mining applications.

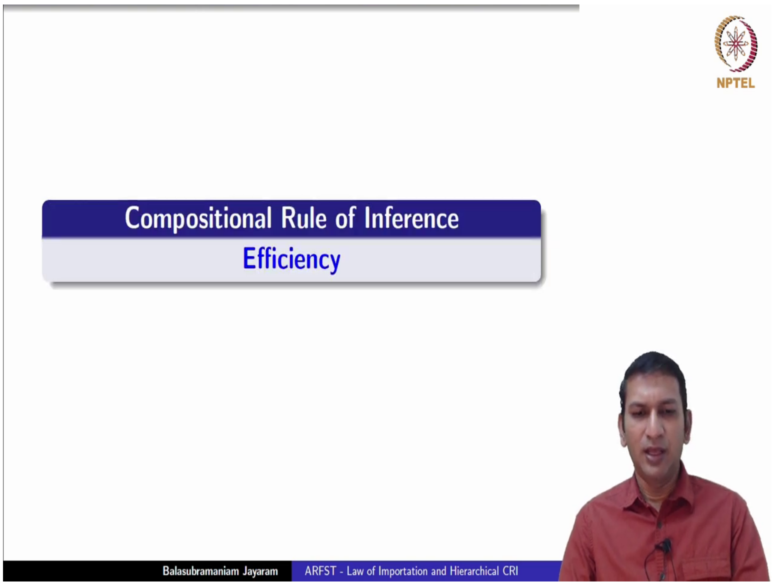
(Refer Slide Time: 06:52)



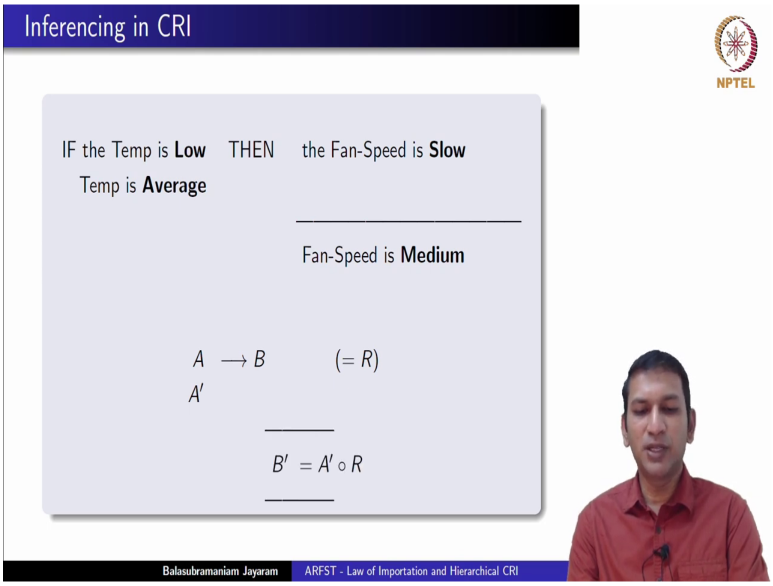
Now, the question is, what about the law of importation? Is it also useful somewhere? Of course we have seen already, in many places where this is playing a role in terms of monotonicity, robustness, interpolativity or continuity; however, now we are asking this question from the point of view of computational efficiency.

We can always take solace in this court of Nikolai Lobachevsky. He said there is no branch of mathematics; however, abstract which may not some day be applied to phenomena of the real world. So, that means, law of importation could also serve some purpose not just in the theoretical or the correctness part of it, maybe also in the computational aspects of a fuzzy inference system. And that is what we will see in the rest of this lecture.

(Refer Slide Time: 07:37)

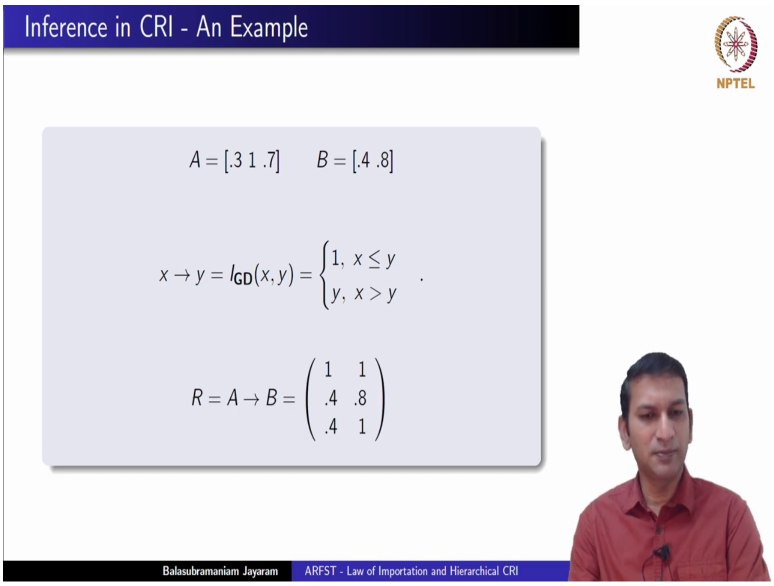


(Refer Slide Time: 07:38)



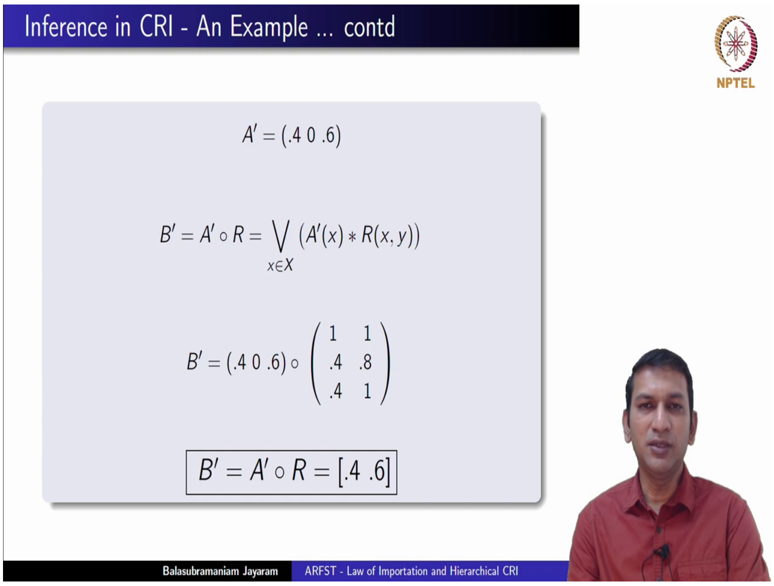
Let us look at inferencing in CRIs. Of course, we have seen this many times over. So, if you have a rule of this form, if the temperature is low then the fan speed is slow and if a given temperature is average, we want to obtain an output fan speed is medium. So, typically the rule is captured in form of a relation and given an A dash which is the input you obtain B dash by composing A dash with R.

(Refer Slide Time: 08:01)



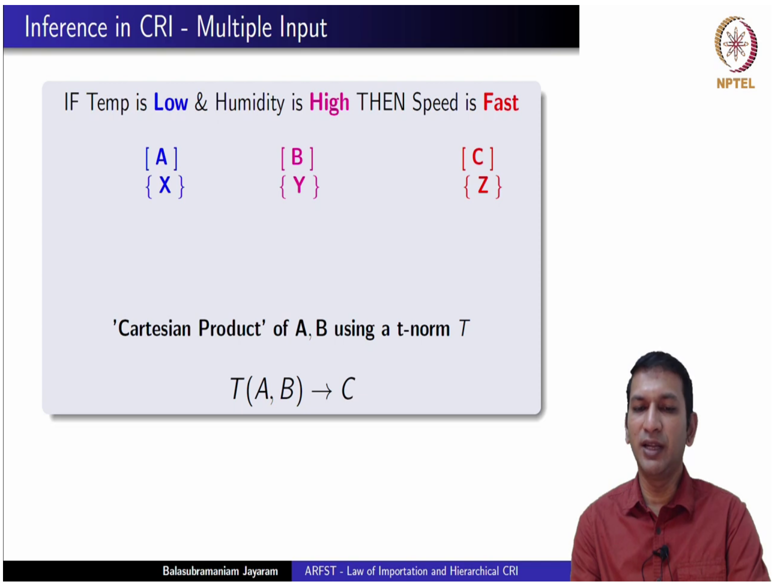
Now, we have seen examples of it before, but once more because it is the numerical aspects of it which will highlight the issue that we are going to discuss now. So, given let us assume that A and B the antecedent and the consequents are given as follows, as these fuzzy sets. We use the girdle implication to relate them. So, the relation would look like this.

(Refer Slide Time: 08:26)



Now, if you are given an input A dash, all we need to do is use the sup-T composition which in this case let us assume some the usual minimum T-norm. So, then what we obtained is this B dash. So, we are using star to be the minimum T-norm.

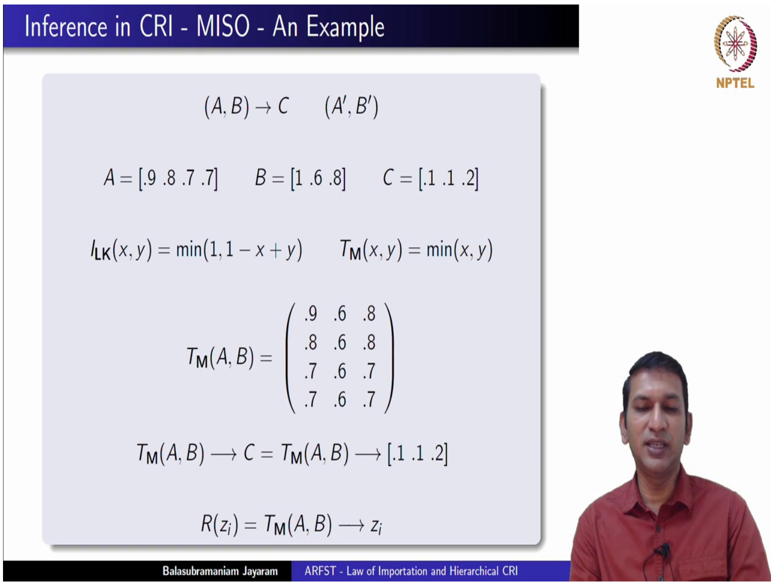
(Refer Slide Time: 08:44)



Now, what happens in the case of multi-input single-output rules? So, let us construct just one of them. We have seen this before too, that we need an antecedent combiner and we do not then deal only with 2-D matrices, but also 3-dimensional matrices if you have two inputs.

So, the temperature is low and the humidity is high, then speed is fast. So, now, this is something like you have a fuzzy set A F of x 1; B from F of x 2 and C from F of y. So, x or we could also take the domain to be x, y and z. So, normally what we do is, we take the cartesian product of A and B using a T-norm and then apply the function F, which is typically an implication to obtain the relation of this particular rule.

(Refer Slide Time: 09:30)

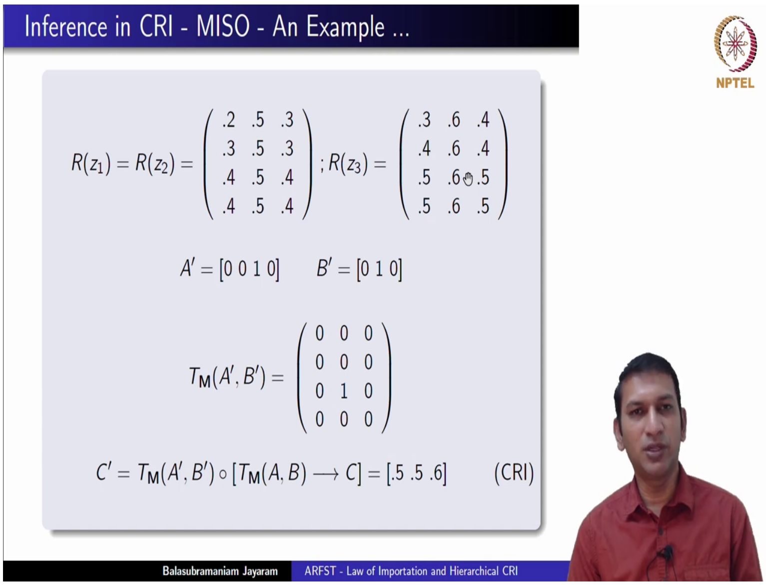


Let us see an example we have seen this before to. How to do this inference? If A is given like this, B and C are given like this. And, let us use the implication for the F to relate the combined antecedent to the single consequents and let us use the minimum T-norm for combining A and B into a single relation.

So, T M of A, B is essentially once again we are using the outer product concepts, 0.9 and 1, it is 0.9, 1 10 0.6, it is 0.9 0.6 0.9 and 0.8 it is 0.8. The rest of the matrix can be filled like this. Now, the input the antecedent themselves are multi-dimensional. So, instead of having a single row vector, we are already having a 2-D matrix. Now, this is what is going to play a role in giving us a relation with respect to the consequents C.

So, we have seen that this typically leads to a 3-dimensional matrix or in this case 3, 2 D matrices because, C has 3 elements. And, we have seen that this can be split down into R of z i for each z i where z i is 0.1, z 1 is 0.1, z 2 is 0.1 again and z 2 z 3 is 0.2.

(Refer Slide Time: 10:47)



So, how do you obtain this? Once again, we have seen this that we take every element of T M of A, B and then use the implication of 0.1. So, 0.9 and 0.1 is we are using the Lukasiewicz implication each one of them and this is how we are obtaining the implication, the values of the matrix.

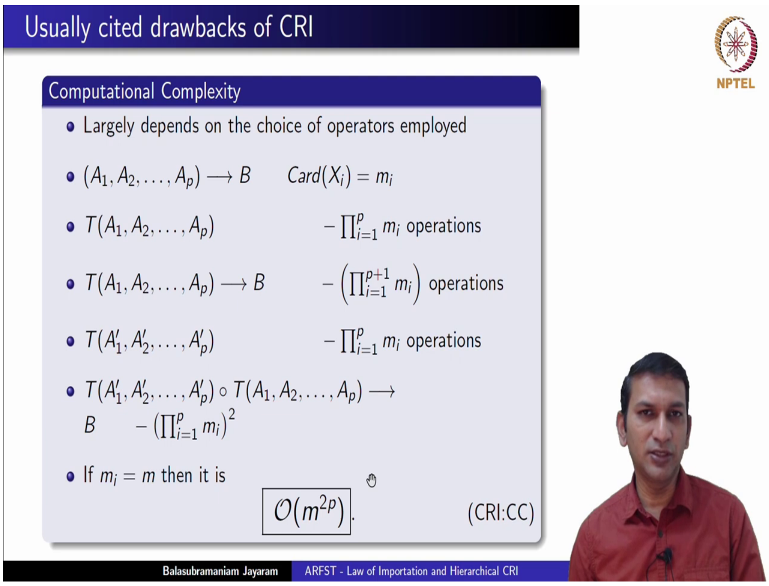
Note that when you use the Lukasiewicz implication. So, this is 1 minus x plus y or 1 minimum of these 2. So, when you take 0.9 and 0.1, it is 1 minus 0.9 plus 0.1 which is 0.2 when we use 0.6 and 0.1 it is 1 minus 0.6 0.1 which is 0.5. So, that is how we obtain this matrix. In the case of, so, z 1 and z 2 are 0.1. So, we obtain the same relation matrix and z 3 is 0.2. So, we obtain a different one. In some sense translated by another 0.1.

So, additional 0.1 comes into picture. Because Lukasiewicz is implication is 1 minus x plus y. Now, so, we have these 3 2-D matrices. Now, given an input A dash B dash, let us assume them to be singleton, then what we do is once again, we combine them using the antecedent combiner which in this case it is a it is a minimum T-norm.

We obtain this and then use this to compose this 2-D matrix with the 3-D matrix. If you remember the analyzer which we have given, we have 3 sheets 2-D sheets on which these matrices are and the steam of A dash B dash is another sheet. So, we keep that matrix on each of the sheets and then in fact do the component wise min and then take the max.

Clearly only this element will come out because the rest of them will be 0. So, 1 here. So, only this component will come out 0.5 0.5 and 0.6. This is what we have seen, right. This is how we do inferencing in CRI when we have a single MISO rule. Clearly, we understand that we have multiple SISO, MISO rules then it is getting even more cumbersome.

(Refer Slide Time: 12:50)



Some of the usually cited drawbacks of CRI follows. Firstly, the computational complexity. Of course, it depends on the choice of operators employed, but let us look at it in the terms of number of operations required. So, now, we have a p dimensional input and a single consequents, single output. If you assume that each of the X i is of cardinal and that is the discretization granularly p consider.

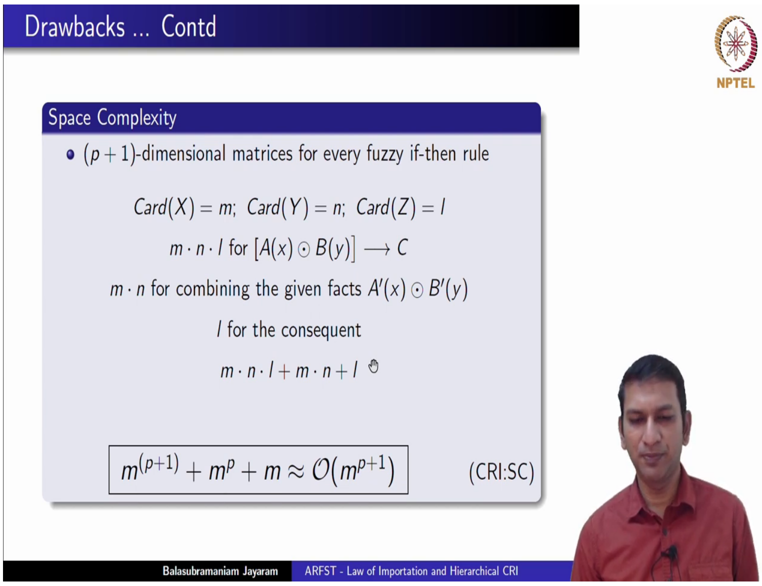
Now, first we need to combine these antecedents. Remember, if we had we in the previous case we had 2-Dimensional inputs A 1, A 2 or A and B. So, we got a 2-D matrix. So, now, if there are m 1 components of A 1 and m 2 components are A 2 then it is m 1 cross m 2. Those many components have to be obtained to make the 2-D matrix. So, if you consider this; that means, we need m means the product of this m 1, m 2, m 3 1 m p.

So, many operations to obtain the matrix itself this is the antecedent. The next to obtain the relation between this combined antecedents and the consequents we need another set. So, total number of operations that are required is this. So, Cartesian product of 1 to p plus 1 such operations. Now, once again for the inference itself, given an input A 1 dash, A 2 dash A p dash. Once, again, these things have to be combined which means these many operations i i is equal to 1 to p and m i operations.

And finally, for the composition itself, we need the square of that operation. So, if you assume that all the m is are same say is equal to m, then essentially what we are looking at is the order is given by the largest of this. So, it is essentially here you have m power p the whole square. So, it is order of m power 2 p.

So, those many operations have to be performed. So, p is the dimensionality m is the discretization, constantly on each domain how many points of discretization that discretization granularity that we are making use of.

(Refer Slide Time: 14:57)

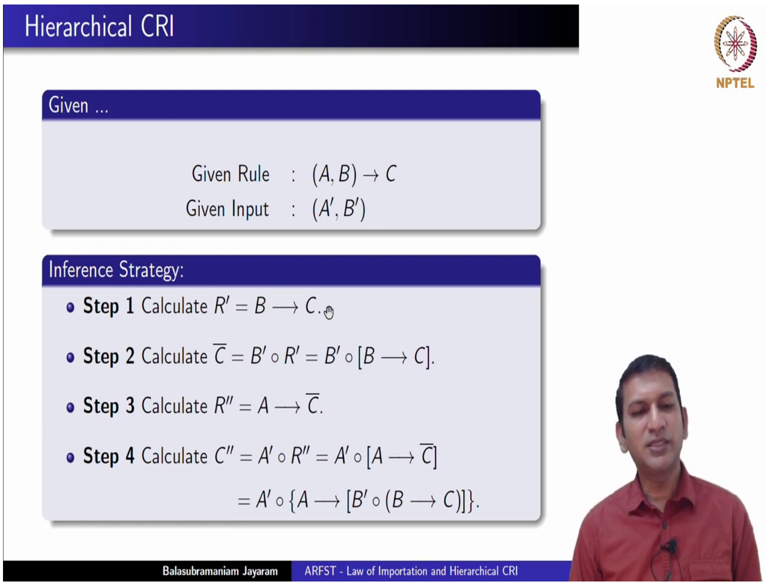


Now, this with respect to the computational complexity. What about the space complexity? How much we need to use as storage? So, we need to store p plus 1 dimensional matrices for every fuzzy if then rule. So, if you assume that the cardinality of X is m and Y n, Z is l. So, for the moment let us consider only two input one output case. So, m n Z and then, so we have seen that to obtain the rule the relation of the rule.

We need a 3-D matrix which will consist of m into n into l those many elements, components and towards doing the inferencing first we need to combine the facts A dash B dash which is again a 2-D matrix with m cross n elements. And, we need l to store the consequents. So, essentially, we need this much to hold; m n l to hold the relation. m n to obtain the inputs and then find the matrix and l to store the consequents.

So, once again assuming that these are actually all same that, m is equal to n is equal to l; then this is for two input and one outputs so; that means, this essentially becomes m power p plus 1 this is m power p and m. So, once again taking the highest order quantity here. In general, this is of order m power p plus 1. So, for computational complexity is order of m power 2 p and space complexity is order of m power p plus 1.

(Refer Slide Time: 16:19)

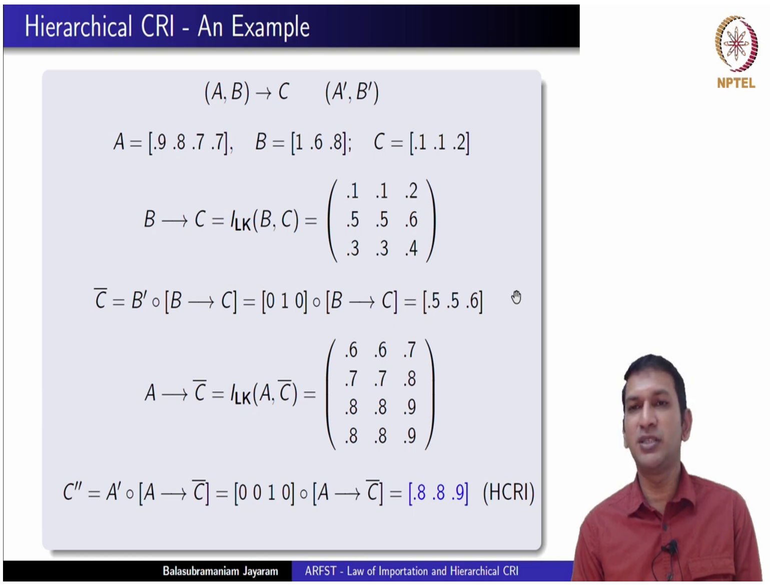


Now, why not perform the operation in a different way? So, look at this hierarchically done CRI. What do we have? We have a rule of the form A, B implies C and we are given input A dash B dash. The hierarchical CRI, it suggest the following procedure. It says, first calculate B implies C, then calculate an intermediate output C bar by using B dash and B implies.

So now, B implies C means it is a single input single output rule, you have a 2-D relation given a B dash you compose it with this R dash and obtain the C bar which is an intermediate output C. Now, the third step what we do is use A and this C bar as the consequents and obtain the rule relation.

Then, obtain the final output by composing A dash with this R double dash. So, overall, this is what you will get. So, at any point of time we are actually only using a single input single output rule, storing that and then given the input we are actually composing and obtaining this and this could also be extended if you have three inputs or n inputs or three inputs.

(Refer Slide Time: 17:30)



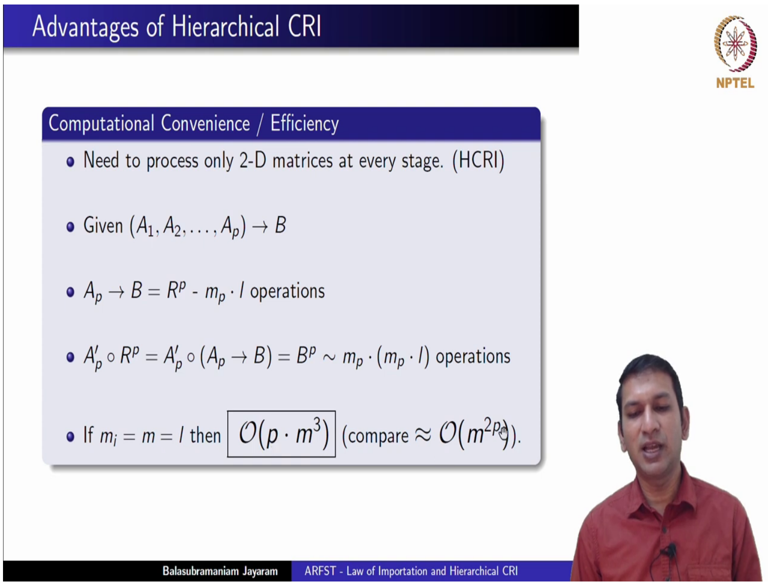
Now, let us calculate this for the same values that we have considered. So, the rule that we have is A, B implies C and A dash B dash of given inputs. Let us assume the same values for A, B and C. So, first what we do is we obtain the relation between B and C only B is the antecedent and C is the consequents. Once again let us use the Lukasiewicz implication. So, the relation that we would get is essentially this.

Now, to obtain the intermediatry intermediary output C bar, it is we need to compose B dash with this B implies C and we know B dash is singleton fuzzified. So, when we applied this what we get is just the middle row which is 0.5 0.5 0.6. Now, this procedure suggests that we should use this C bar as the current consequents and use A as the antecedent and for this rule A implies C bar, we need to obtain the relation of the rule.

Once again using Lukasiewicz implication, this is the relation you will get and the final step says that take A dash the given input and compose it with this particular relation R double dash and obtain your final output. So, once again A dash is singleton fuzzified which means, it has a one only at the third element which means the third row will come. So, this is essentially how we proceed. It is clear that, we were able to do it in a single slide.

Which means it is much easier than what we have seen in CRI, even for a multi-input single-output case, not only that there are many advantages.

(Refer Slide Time: 18:58)

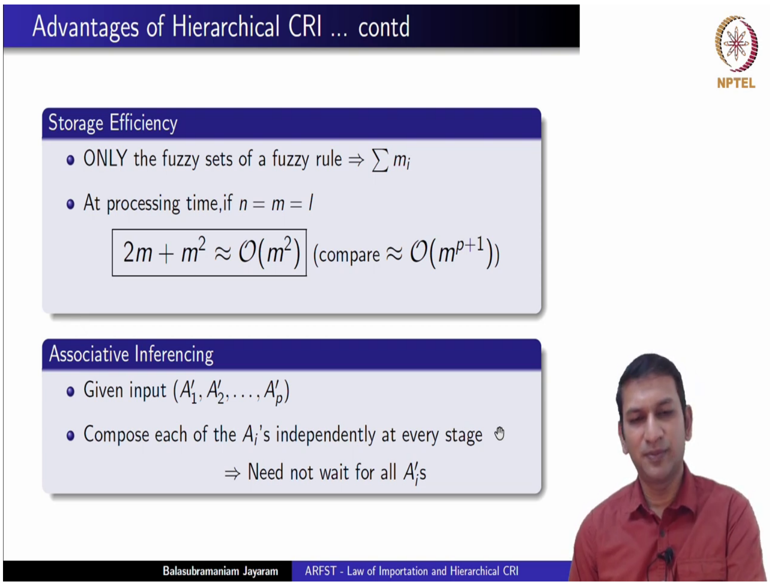


The first advantage when you see is we need to process only 2-D matrices at every stage; because we are only considering one consequents of antecedent. So, even if you have p dimensional antecedents at any point of time, we are only taking A p with B we get an R p then we compose A dash with this R p and get a new say C 1 then, we will take A p minus 1 and then that B 1 or C 1 and then compose between A r obtain consider them as a antecedent the consequents and get the relation.

And then given A p minus one dash as the input we compose it and we go on. So on and so forth. So, essentially to obtain this single input single output I mean single input and the single output case we need only m p into l operations. Now, to compose once again, this is what it means. So, essentially it is m p into m p into l operations note that at any point of time we are going to only consider single input single output rules and a single input A dash or A p dash and obtain the corresponding consequents.

So, essentially all the time we are only using m cube operations and we may have to do it p times. So, the overall computational complexity is nothing but p times m cube compare this with the usual CRI which is m power 2 p. So, it is clear that m cube is the one that is going to dominate depending on the granularity. So, essentially it is m cube p of it is a order of m where has here it is actually going to increase exponential.

(Refer Slide Time: 20:33)

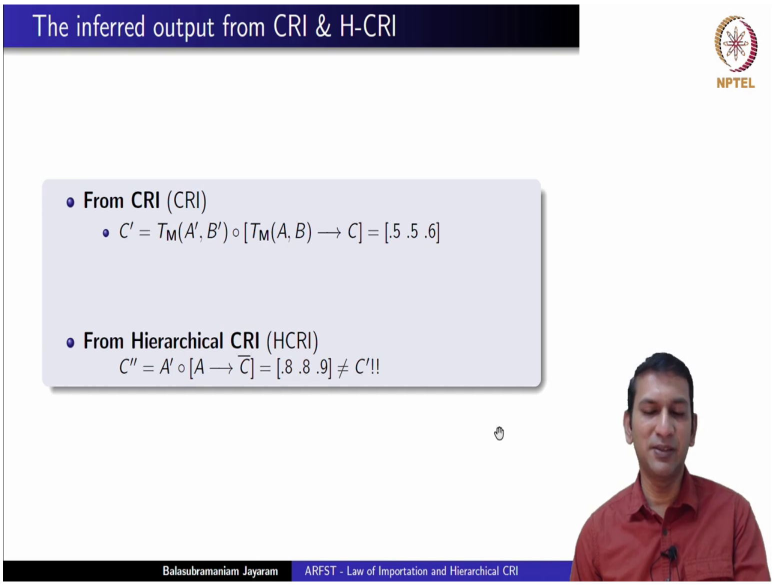


What about the storage efficiency? You notice that at any point of time, we are only storing the fuzzy sets of a rule we are not storing the rule relation at all. So, that means, if each of the domains there is m i discretization with the number of points of discretization m i essentially sigma m i is what you need to store. And, at processing time of course, we need a relation and the given input and the given output.

So, essentially it is 2 m plus m square. So, order of m square is the highest thing that you will take. Compare this with order of m power p plus 1. So, that is the difference in terms of storage is almost m square always that is m power p plus 1. Quickly it saturates exponential; not only computational storage deficiency are available, we also have associative inference.

For instance, if you are given this input, we can compose each of the A is independently at every stage which means we need not wait for all the inputs A A dash and also not necessarily in that order. So, these are some of the advantages of having hierarchical CRI.

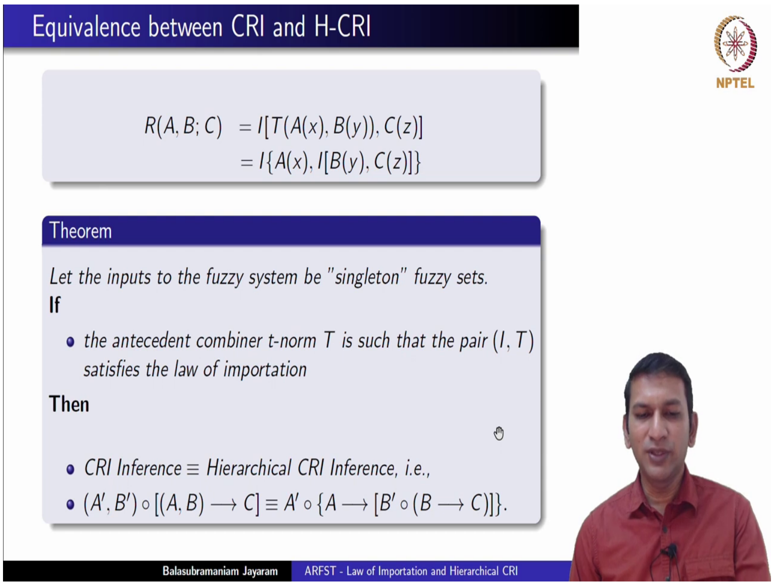
(Refer Slide Time: 21:39)



But, what about the inferred outputs from the original CRI this hierarchical CRI? Even when we have the same A, B as Bs and Cs and the same operations. We noticed with from CRI the output we got was 0.5 0.5 0.6 whereas from the hierarchical CRI it was 0.8 0.9 0.9 which is not equal to the original inference. Well, is it a problem?

Well, yes, because we know that CRI, when we consider some operations, we have discussed at length about it about the desirable properties CRI in terms of interpolativity, monotonicity, robustness and continuity by moving to this form of an inference procedure. We should not lose them. So, while we have a computationally efficient procedure, we should not lose the accuracy, efficiency cannot hit efficacy.

(Refer Slide Time: 22:34)

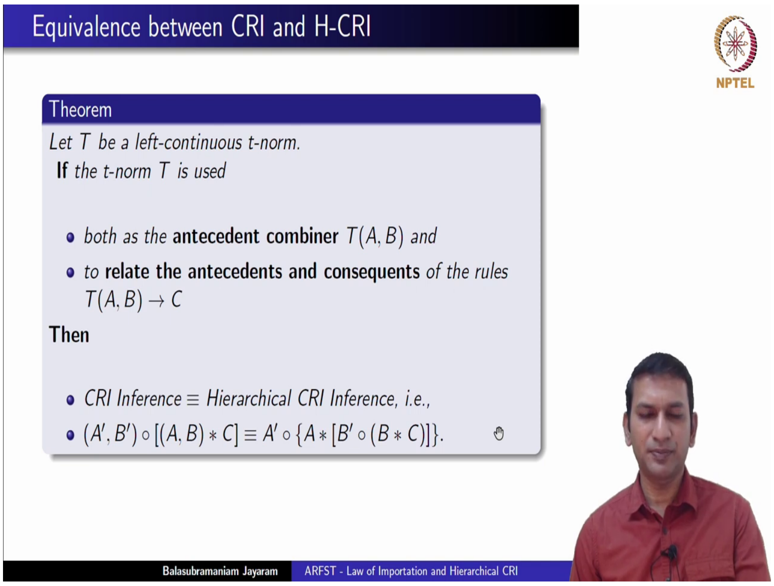


Now, is there an equivalence? Can we show that the outputs obtained from these two will be equivalent? Well, yes, now for that look at the rule itself. So, consider the two inputs single output. So, to obtain the relation what we do is, we first combine A and B with the T and then apply an implication on this combined A, B to C.

So, in some sense, this is how it looks like for any given triple x y z. But, now, what hierarchical CRI does is converts it into a sequence of implications. This is how it is seeing. Now, clearly these two are equal if I and T enjoy law of importation. So, if we have a result which says that if the inputs to the fuzzy system are singleton fuzzy sets and if the antecedent combiner T-norm T is such that the pair I, T satisfies the law of importation.

Then the inference obtained from the CRI is exactly identical to the one obtained from hierarchical CRI and what if the inputs are not singleton?

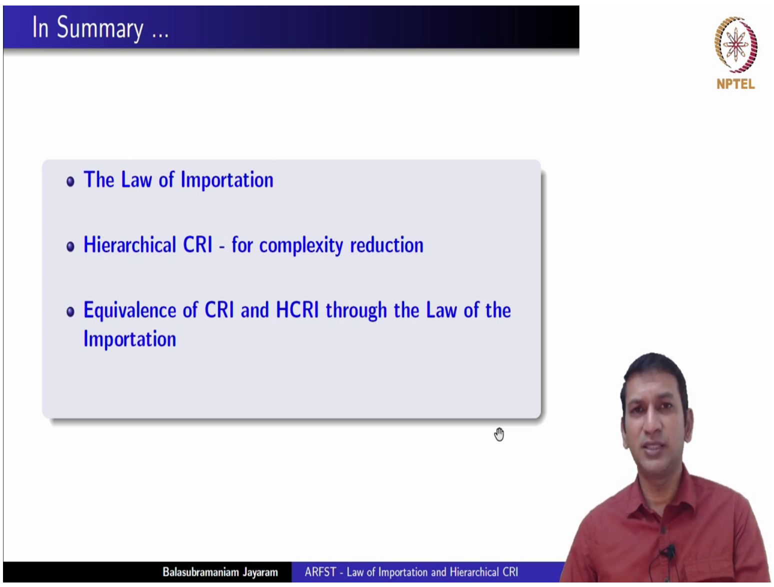
(Refer Slide Time: 23:40)



Well, then, essentially by using the associativity of T-norm which is essentially like the law of importation with I substitute T. We see that we could use the same T-norm as antecedent combiner and also to relate the antecedent and consequents which is more like a R check rule.

We see that CRI and hierarchical CRI inferences are in fact, equivalent. So, we see the important role played by law of competition here in ensuring that the computationally efficient mechanism is in fact, also correct.

(Refer Slide Time: 24:16)



In summary, we have studied law of importation in one of the previous lectures. And, we know that there are many families of fuzzy implications which do satisfy this law of importation. We have seen that in the case of residuated lattices it is always available and we have made use of this multiple times improving desirable properties of FRIs especially.

But, in this lecture, because CRI was computationally pretty inefficient, we moved to the hierarchical version of CRI for complexity reduction. However, we found that the outputs obtained from the hierarchical CRI were not equivalent to that of CRI. And, that is where we felt we found that the law of importation comes to a rescue and says, ok.

If the operations involved satisfy this property then, we can ensure the output obtained from CRI is the same as that of what is obtained from an hierarchical CRI (Refer Time: 25:13).