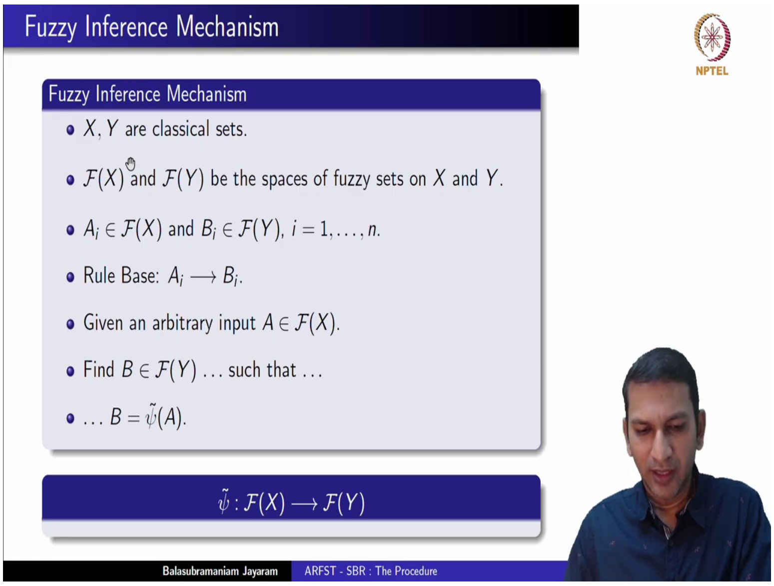
Approximate Reasoning using Fuzzy Set Theory
Prof. Balasubramaniam Jayaram
Department of Mathematics
Indian Institute of Technology, Hyderabad

Lecture - 34
Similarity Based Reasoning

Hello and welcome to the second of the lectures in week 7 of this course titled Approximate Reasoning using Fuzzy Set Theory, a course offered over the NPTEL platform. In the previous lecture, we looked at Similarity Based Reasoning using a visual illustration of the entire process. In this lecture, we will look into the details, the operations involved, the steps involved in performing this inference, similarity based in reasoning inference. We will also see how it entails some restrictions on the rule base being considered.

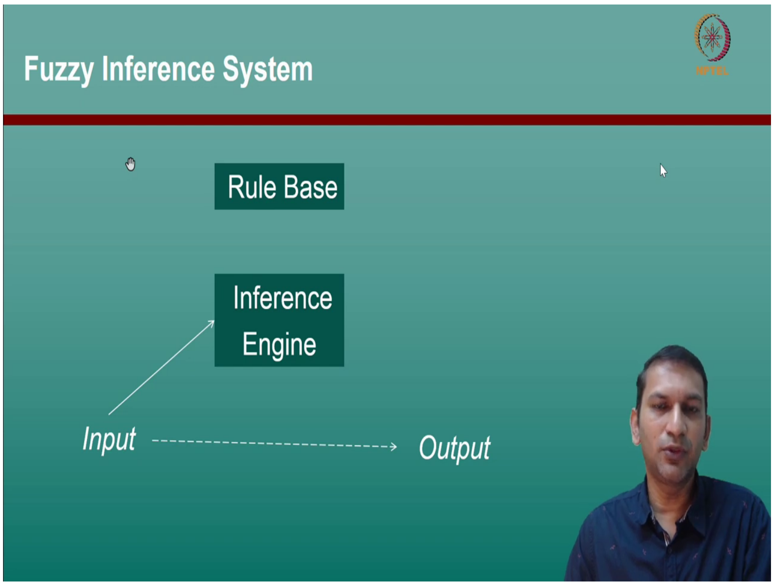
(Refer Slide Time: 01:00)



So, if we know what a fuzzy inference mechanism is. We have this input and output spaces X and Y. We consider fuzzy sets coming from these two denoted as F(X) and F(Y). The antecedents A i and the consequence B i come from the fuzzy spaces or are fuzzy sets on the corresponding spaces X and Y. And, we have a rule base which relates the antecedents to the consequence and what we need is given an arbitrary input A which is a fuzzy set over X, we need to find the B which is the fuzzy set over Y such that B is related to A.

And, this we do not do it in vacuum, for us the ground truth that is available is in the form of these fuzzy if then rules. So, we have seen that fuzzy inference mechanism can be thought of as a function from the set of fuzzy sets X to the set of fuzzy sets Y.

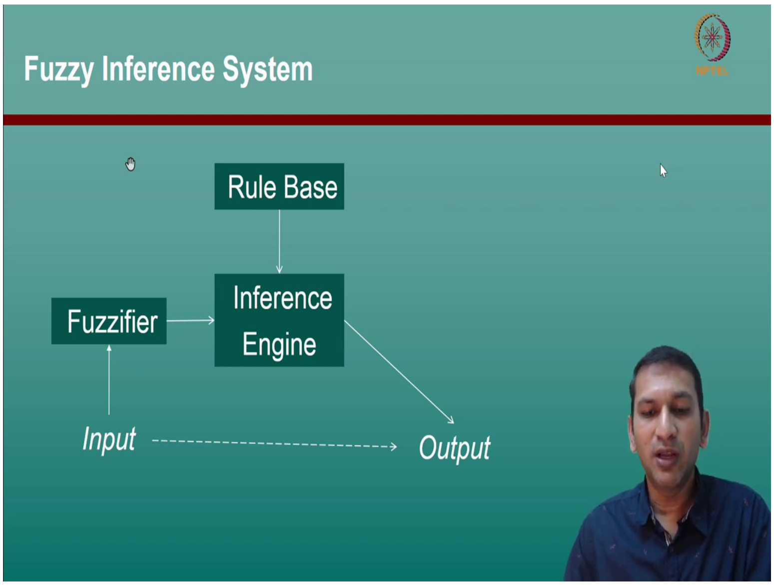
(Refer Slide Time: 02:01)



In the previous lecture, we looked at diagrammatic schema of the fuzzy inference itself. What we would like to do is map a given input output, if you remember the example of an air conditioner, the control system for an air conditioner; the input is the temperature and output is the fan speed. Now, this is the mapping that we would like to do. How do we like to do this?

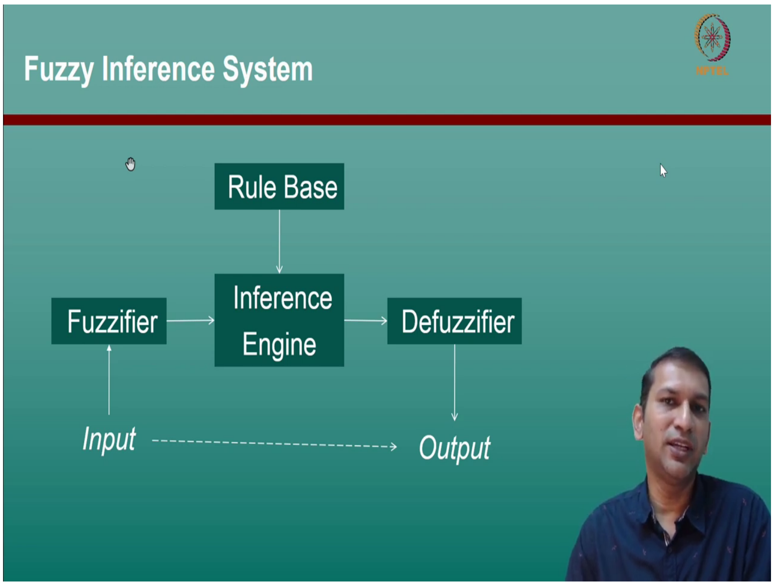
With respect to the given knowledge which is contained in the rule base. For this we need an inference engine. We give this input to this inference engine and it discusses, takes the help of the rule base and coming out with an output.

(Refer Slide Time: 02:50)



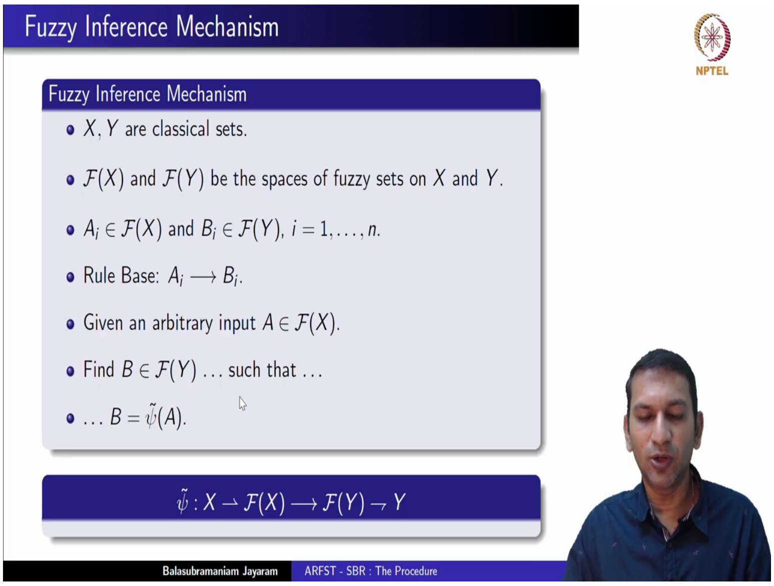
But, remember largely we are discussing with inference engines which can handle fuzzy sets. So, often it is there is a need to fuzzify the given input to a fuzzy set over X and then feed it to the inference engine. The inference engine then discusses with the rule base and gives us an output. Often, either it could be a direct output in the form of a real number or to the output space itself.

(Refer Slide Time: 03:18)



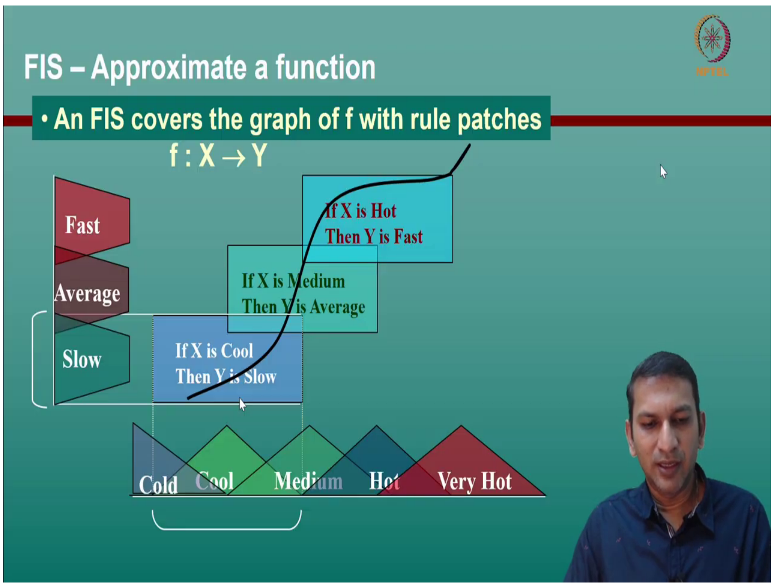
Or it might go through another process step, it might output a fuzzy set over Y which needs to be suitably defuzzify to get a value in Y. So, the overall function that the fuzzy inference system tries to capture is a mapping between the input and output spaces; that means, it is trying to capture the function F from X to Y.

(Refer Slide Time: 03:45)



So, now this means the fuzzy inference mechanism can itself can be seen not just as a function from F(X) to F(Y). But as a overall function moving from X to F(X), lifting the given value to a fuzzy set on X, mapping it to a fuzzy set on Y and then defuzzifying it, mapping it to a value in Y.

(Refer Slide Time: 04:01)



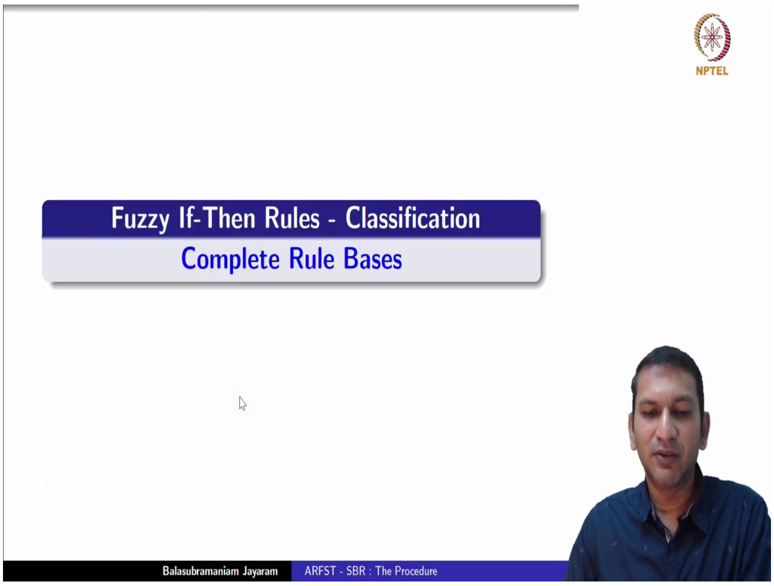
Now, we also know that the fuzzy inference system what it does is approximates a function. We have seen this, given an f from X to Y, this is what we call the system function. The function that is inherent in the system, how the system operates which mostly we do not know. We have seen one particular example, so let us revisit that. So, assuming this is the function that we are trying to capture. This is the functioning, the ideal functioning of a control system that is fitted to an air conditioner. This is what we have done in the previous lecture.

We have taken these are the fuzzy sets as on the input and output domains and we have tried to relate these in the form of rules. So, we saw that if the rule is given as if X is Cool then Y is Slow, it does two things. Firstly, it essentially captures some local knowledge about this part of the domain. It is clear that only points that fall within the support of X are able to excite this fuzzy set Cool; that means, this particular rule is largely responsible and only responsible for this local neighborhood of the domain.

Similarly, the other rules also are capturing some local knowledge about some part of the domain. This is one interpretation or one perspective. The second thing that we have seen is an FIS, a fuzzy inference system covers the graph of f with overlapping rule patches that is what we have seen. So, every rule what it does it will captures some part of the domain, I mean some knowledge about some part of the domain and this is how they are stitched together to approximate this function.

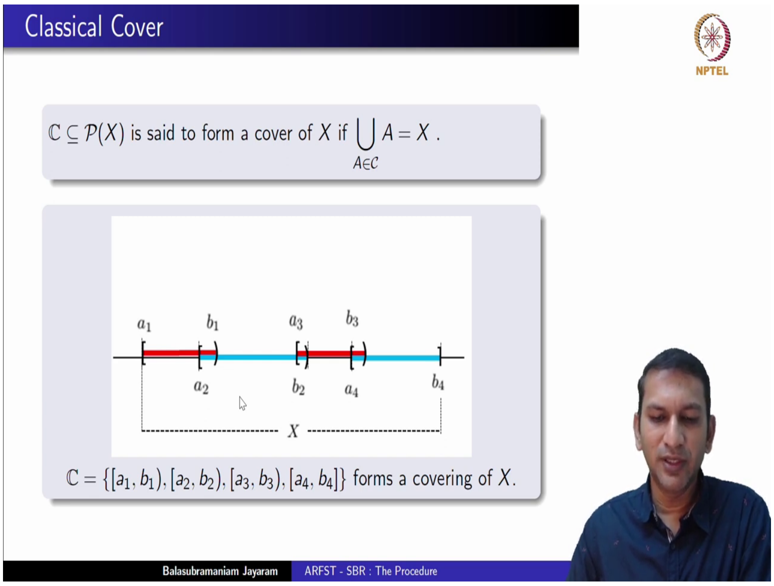
Now, this means we need to give special attention to the rule base itself and also to the antecedents and the consequence. So, it means we need to carefully choose these antecedents.

(Refer Slide Time: 06:10)



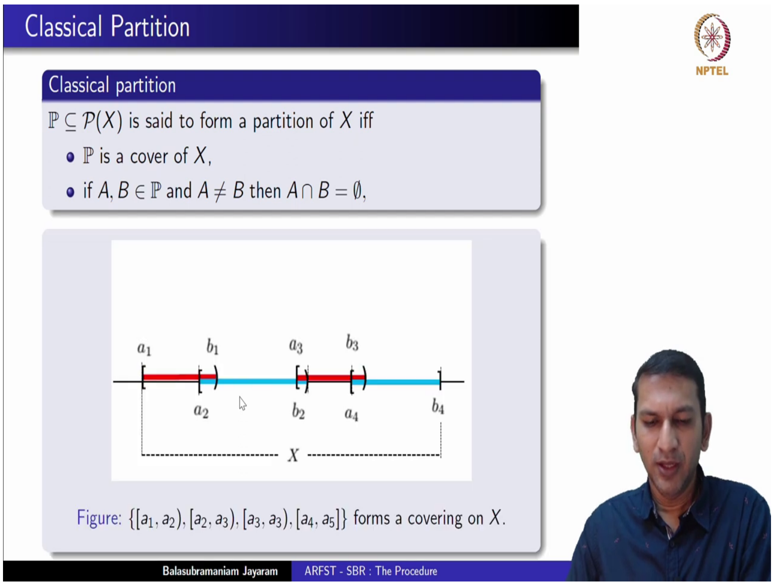
And, one such way to choose is to have a complete rule base. It could also be seen as yet another classification of fuzzy if then rules themselves, a set of rules.

(Refer Slide Time: 06:23)



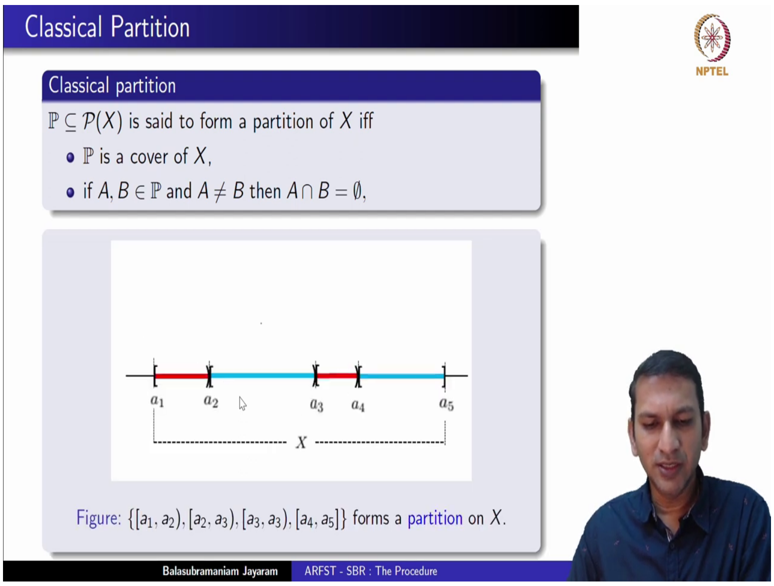
Let us quickly recall some of the concepts that we have introduced, perhaps in the very first week itself. We understand what a classical cover of a set is, given an X a collection of subsets of X is said to form a cover if their union contains X. For instance, if X is this interval a 1, b 4 then these 4 intervals form a covering of X.

(Refer Slide Time: 06:49)



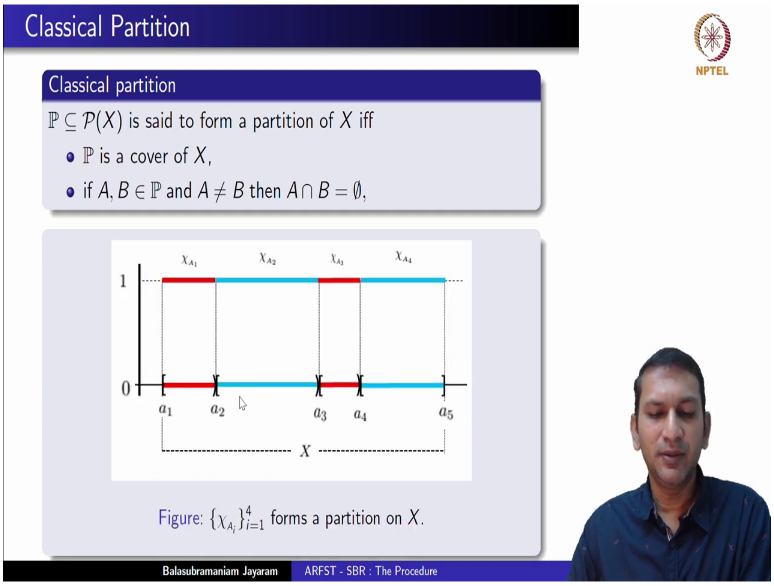
But, this is not a partition. For a partition, we not only needed to cover X, but we want that the pieces of the partition do not overlap. That means, any two subsets from this partition, from this cover should not overlap, their intersection should be empty.

(Refer Slide Time: 07:08)



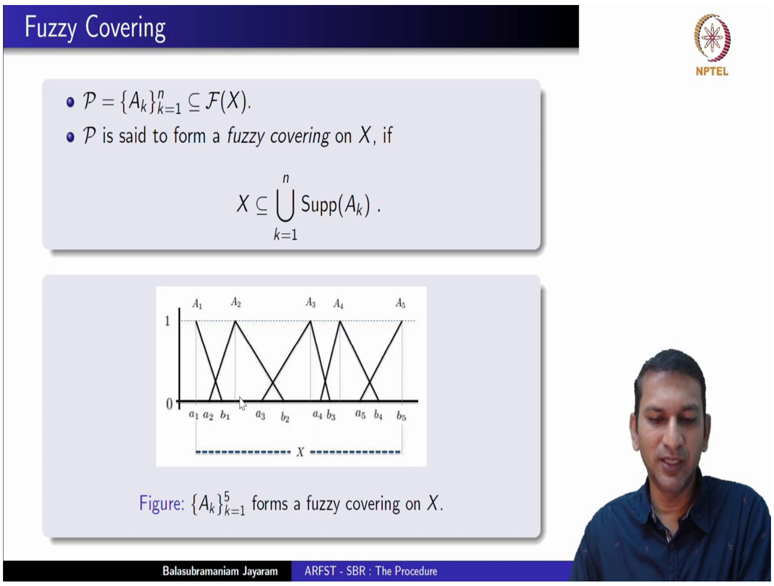
So, this is the covering, but instead if we consider these four intervals, they form a partition.

(Refer Slide Time: 07:12)



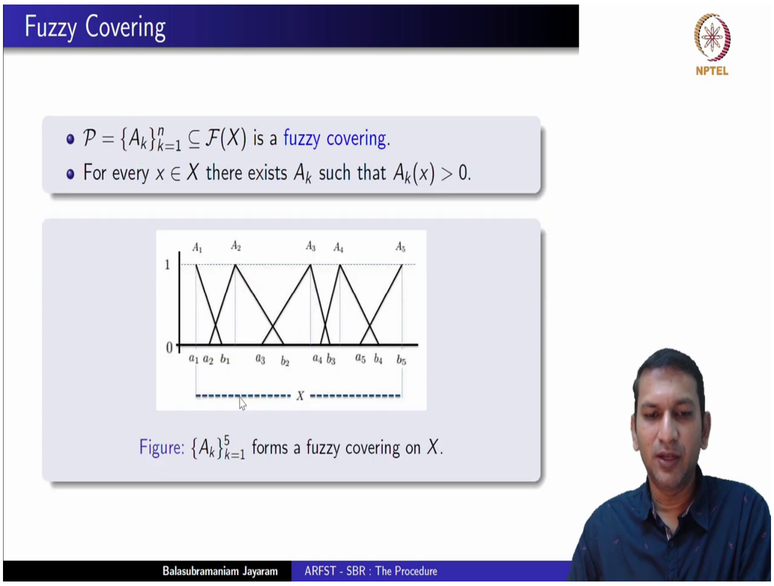
It is from here we have generalized to the case of fuzzy sets. So, looking at these intervals as characteristic functions, we saw that it could be generalized similarly, but now with the added advantage of having overlapping fuzzy sets.

(Refer Slide Time: 07:29)



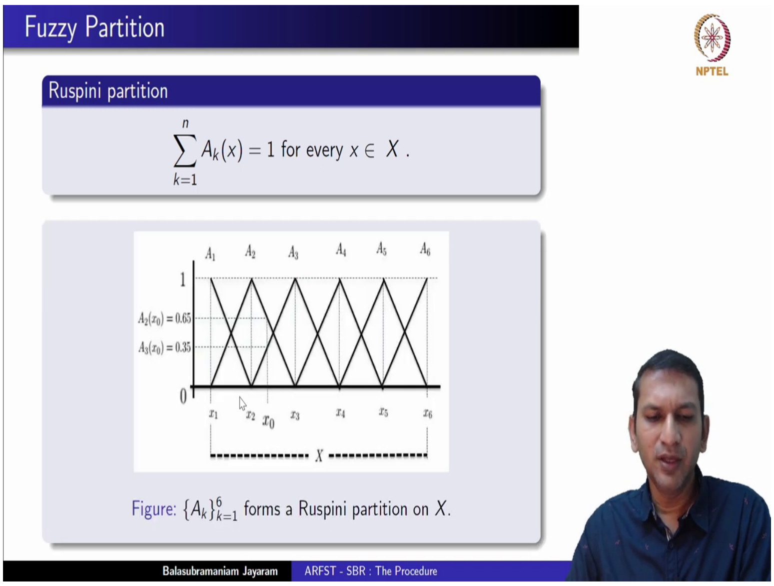
What is a fuzzy covering? It is once again a collection of subsets, but fuzzy subsets of X such a collection is said to form a covering on X, if the union of it supports the union of the support of its members contains X. For instance, if we consider the X to be between a 1 and b 5, the interval a 1, b 5; it is clear that these 5 fuzzy sets form a covering of X.

(Refer Slide Time: 07:56)



Now, it could also be equivalently written like this, a collection of fuzzy sets on X, the form of coloring if for every element in the domain for every x in X there is some fuzzy set to which it belongs to non-zero membership value; that means, it belongs to some fuzzy set to a degree greater than 0.

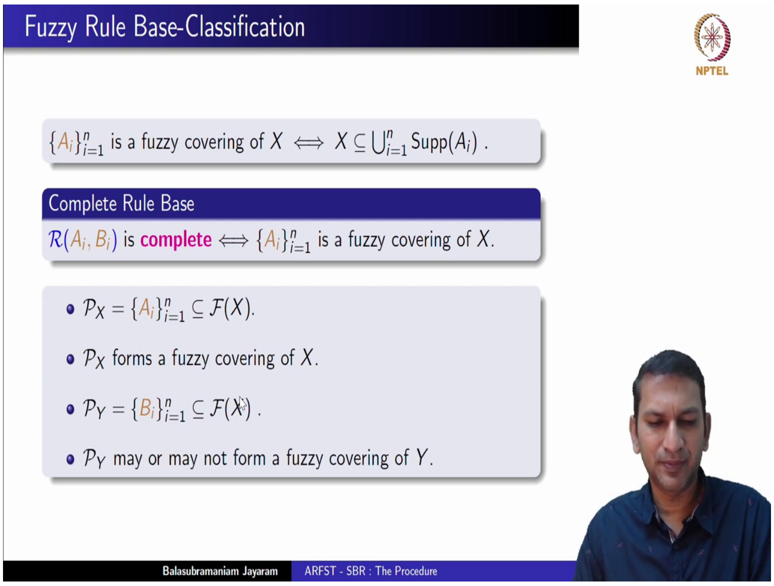
(Refer Slide Time: 08:18)



Well, we have seen fuzzy covering is different from a fuzzy partition. This is how literature, it is interpreted as. One particular partition that has been found extremely useful is that of Ruspini partition which says that and if you are then the collection of fuzzy sets that you have, they should first of all form a cover. And secondly, every element of x, it can belong to more than one member of the collection.

But, the overall membership degrees, the sum of the membership degrees to which x belongs to these fuzzy sets should be equal to 1. The moment you put this equation, it automatically implies that for every x there exists some k, such that A k of x is greater than 0 which means the collection of fuzzy sets should also form a cover. So, we have seen this example earlier. So, this forms a Ruspini partition on X.

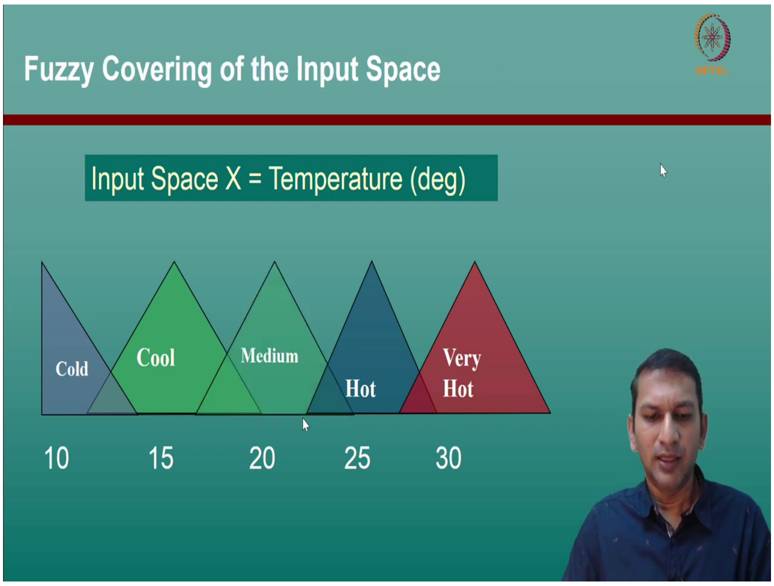
(Refer Slide Time: 09:20)



Well so, now if we have a collection of A i's, we know that it forms a covering fuzzy covering of X, if and only if the support the union of the support of A i's contains X. Now, let us define what the complete rule base is, if you are given a set of fuzzy if then rules, the rule base we say it is complete if and only if you pick up all the antecedents, collect the antecedents and put them together.

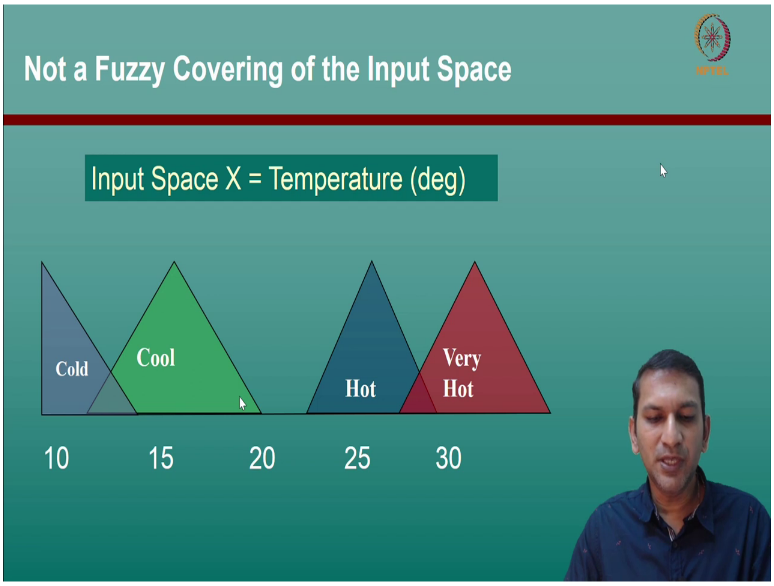
This collection should be a fuzzy covering of X, that is when we say that this rule base is complete. So, now, earlier we were picking antecedents from the fuzzy sets on X, but now we need to be careful in our choosing. So, we will denote by P X, a collection of sets fuzzy sets in X which form a covering fuzzy covering of X. Similarly, by P Y denote a collection of fuzzy sets on F of Y which may or may not form a fuzzy covering of Y.

(Refer Slide Time: 10:37)



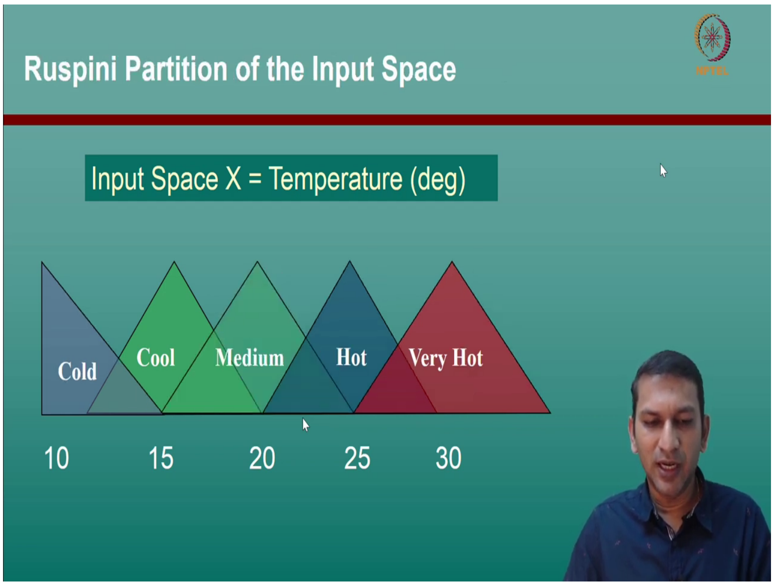
Let us go back and see how this impinges on the collection that we choose. For instance, we have seen that in the previous lecture that we have considered these 5 fuzzy sets. Now, clearly they form a fuzzy covering of the input space.

(Refer Slide Time: 10:47)



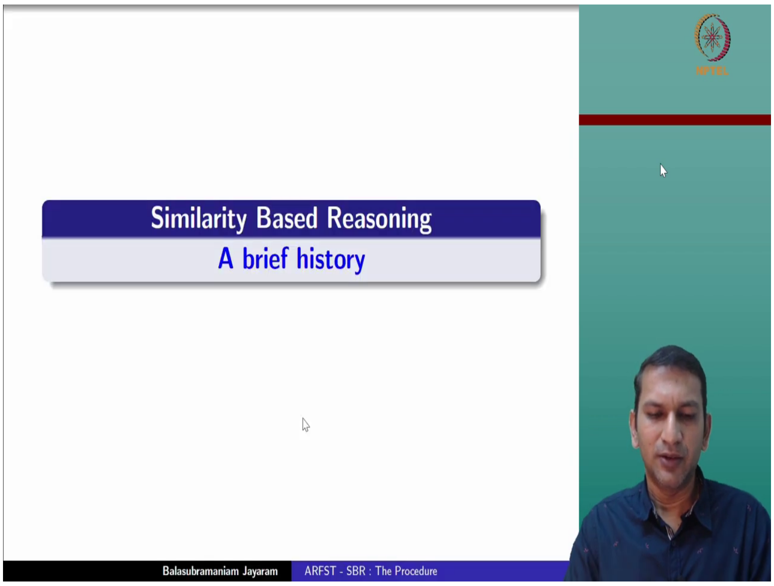
If this piece were missing, then it would not be a fuzzy covering of the input space. Because, when you have an element falling with between say 20 and 22, you would not have any rule being exited and for that input fuzzy inference system will not be able to come up with the output.

(Refer Slide Time: 11:15)



So, this is the fuzzy covering of the input space and, you will immediately recognize from the shapes that this set of functions, this collection of fuzzy sets they not only form a cover, but they also form a Ruspini partition of the input space.

(Refer Slide Time: 11:27)

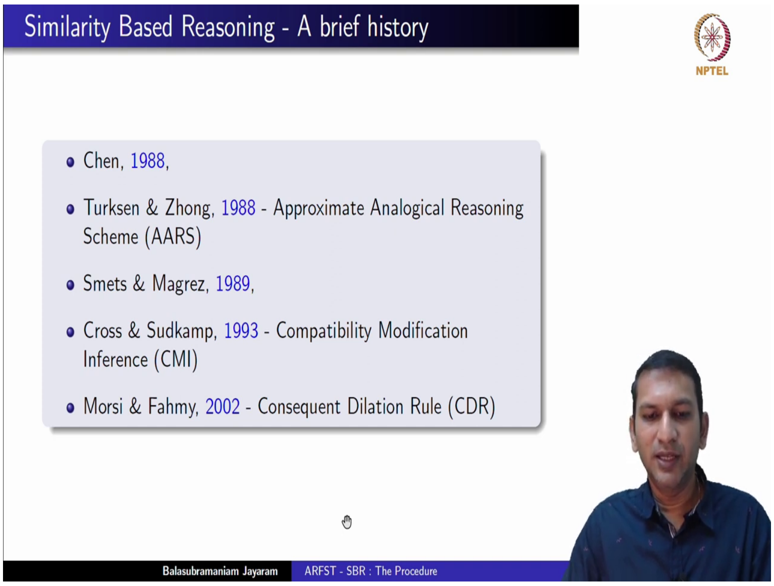


With this let us move on to looking at a brief history of similarity based reasoning.

(Refer Slide Time: 11:33)

In the previous lecture, we have seen that the earliest practitioners, if you look into the literature you could trace it back to two people. The first of them is Ebrahim Mamdani, who in the mid-70s, over period of few years, different works proposed what we now call as the Mamdani fuzzy system. And, almost a decade later came Professor Michio Sugeno, who proposed another way of a fuzzy system, inferencing using fuzzy sets which is called the TSK fuzzy system, the Takagi Sugeno Kang fuzzy system.

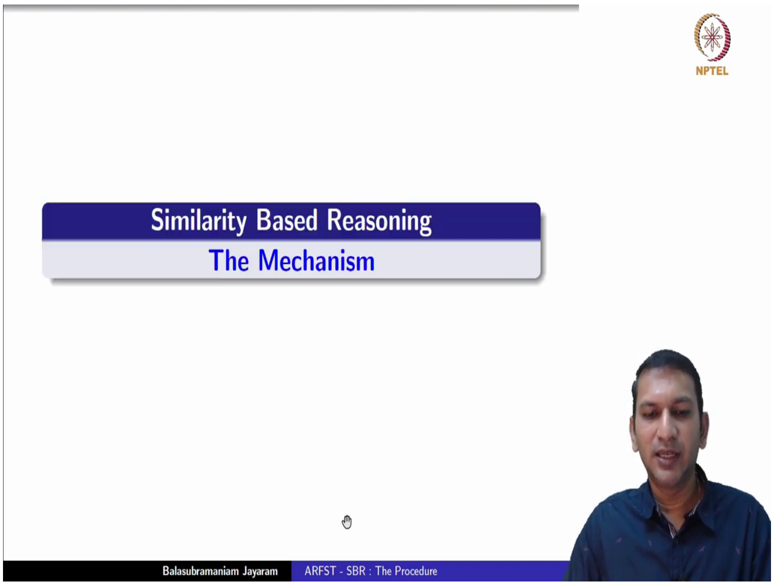
(Refer Slide Time: 12:18)



But, these were not the only people, you will see that later on there were many people who had proposed specific types of fuzzy inference systems. Chen in 1988, Turksen and Zhong, they proposed a fuzzy inference system called the Approximate Analogical Reasoning Scheme. Smets, Magrez in 1989 proposed another such scheme. Cross and Sudkamp they proposed another fuzzy inference system in 1993 called the Compatibility Modification Inference and Morsi and Fahmy in 2002 proposed what they call the Consequent Dilation Rule.

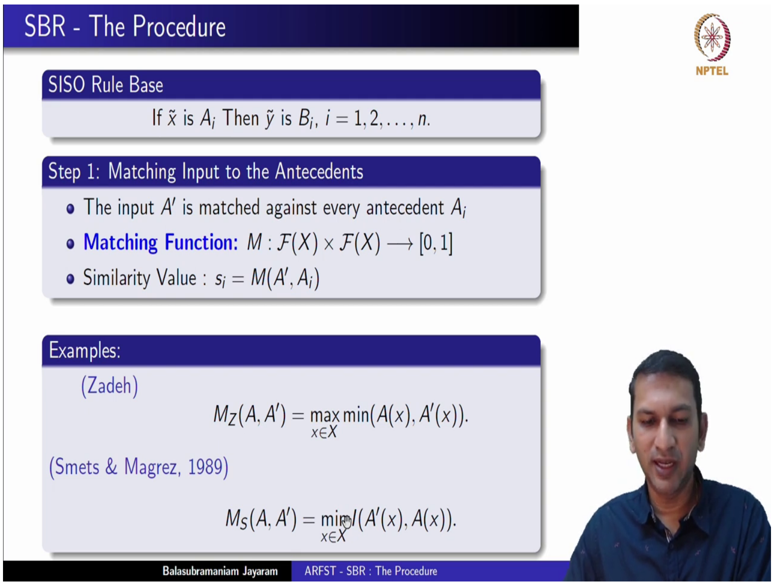
But, get easily be seen that these are essentially some specific cases of similarity based reasoning.

(Refer Slide Time: 12:57)



Now, let us get into the mechanism itself.

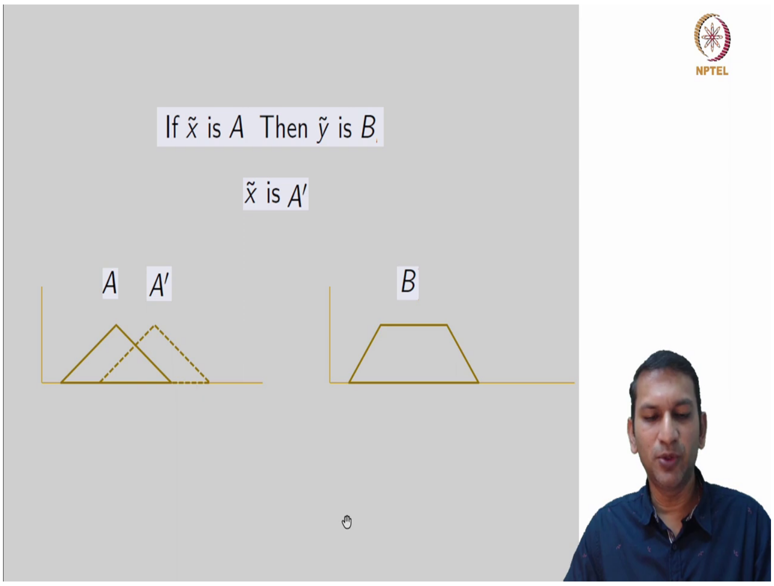
(Refer Slide Time: 13:01)



Once again to begin with, we have a single input single output rule base. If x tilde is A i then y tilde is B i, we have n such rules. Now, what is step 1? We are given an input A dash, we need a B dash that is the output. The first step is to match this input A dash against every antecedent A i, towards helping us in this we employ a matching function which will denote the rest of the lecture series by M.

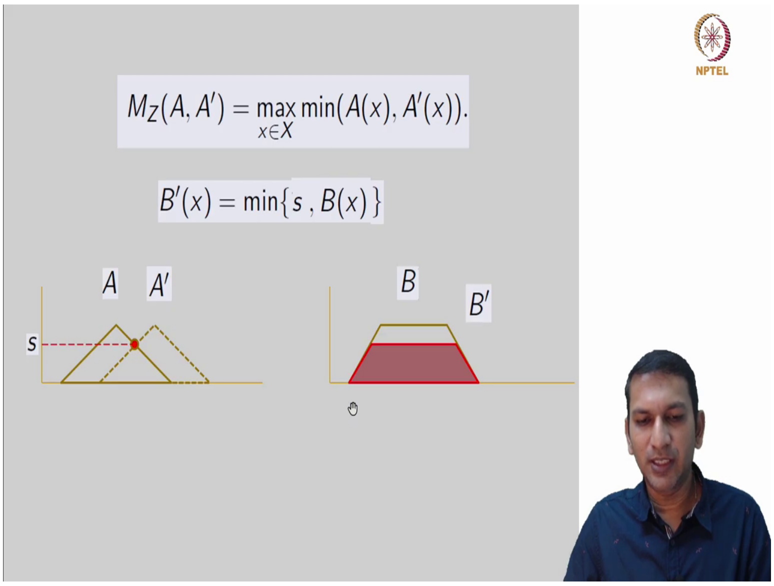
What is this function? It is a function from F(X) cross F(X) to [0,1]. It takes the input A dash and matches it against every antecedent A i, these A i's are fuzzy sets on X. So, this matching function takes these two fuzzy sets on X and gives us a value in the interval [0,1]. And, this is what we call a similarity value and we will indicate it by s i, where s i is the similarity value between A dash and A i as measured by M.

(Refer Slide Time: 14:18)



Let us look at a couple of examples of such matching functions. The first of them has been proposed by Zadeh himself and the second one by Smets and Magrez. So, let us look at how this matching function looks, like in a particular case. Let us take a single rule x tilde as A, then y tilde as B. Let us assume these are the fuzzy sets given to us, A is a triangular fuzzy set on x, B is a trapezoidal fuzzy set on y. We are given the input A dash, let this be the input A dash.

(Refer Slide Time: 14:42)

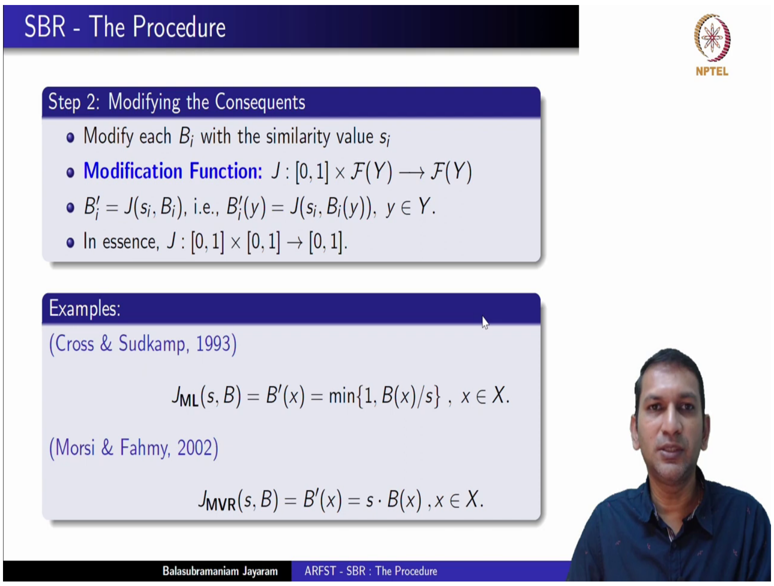


Now, what we now want to do is as step 1, we want to find the similarity between A and A dash. If you use this Zadeh's matching function, what would the similarity be for the A and A dash that we are considering here? Well, we need to apply this formula, visually how would it look like? Look at this, this is essentially applying the minimum t norm on these two sets A and A dash.

So, now, as you vary x over entire domain x, this is what it would be. At point wise we are taking the minimum; so, we can clearly see here it is 0, here it is 0. So, this is again going to be a fuzzy set, that is what is indicated here whose support will essentially be the intersection of the supports of A and A dash because, of the operation minimum essentially for any t norm here. So, this is what is going to give you this part of the formula which is finding the minimum of A x comma A dash x as x varies over the entire domain X.

Then, we need to apply the maximum of this, essentially taking the supremum of this and that is essentially this point. So, now, this is essentially the similarity value is that we have. So, this is what we have found out. So, the first step is finding out the similarity between A given A dash and the antecedent of a rule. We have seen for only one rule.

(Refer Slide Time: 16:03)



Now, what is step 2? Step 2 is using the similarity value, we modify the corresponding consequent. Each rule has A i and B i, the antecedent and the consequent. We have matched the A dash with A i and found the similarity value s i, using this s i we are going to modify the corresponding consequent B i. And, for this we will take the help of a modification function which we will denote by J.

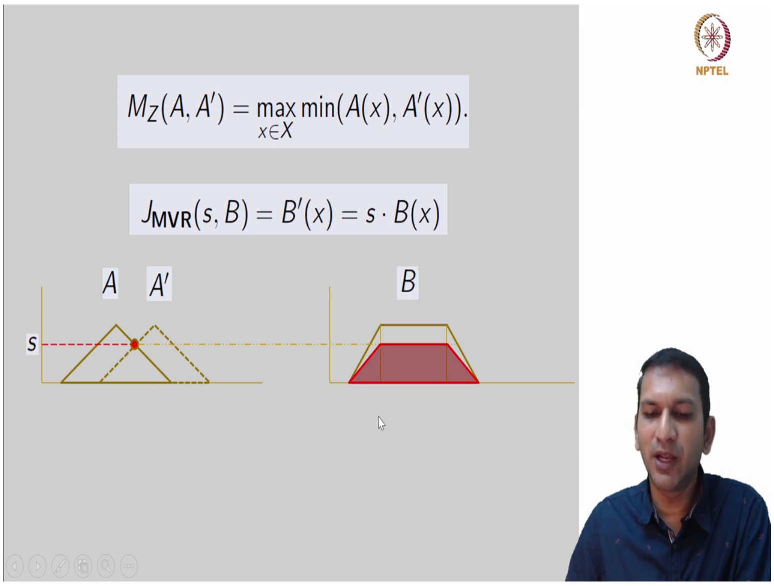
Note, that this is the function from [0,1] cross F of Y to F of Y. It takes the similarity value s i which is an element of the interval [0,1]; takes B i which is the fuzzy set on Y and gives us a fuzzy set on Y. Gives you modified fuzzy set on Y which well denote by B i dash. So, B i dash is essentially J of s i comma B i. So, B i dash is a fuzzy set on y; that means, B i dash will take values for each of the y in the domain of y.

It can be represented like this B dash of y is J of s i comma B i of y. But, notice one thing; so, J takes two values, it is a binary function, s i comes from [0,1]. And, we said that essentially it is acting on fuzzy set B i, but acting on B i means essentially acting on the membership values taken by B i over y. So, this is also a value from 0 to 1. So, essentially we can use any binary function on [0,1] which means we could use any fuzzy logic connective.

Now, let us look at some of the examples of modification function that are being proposed in the literature. Earlier, we saw the matching function proposed by Smets, Magrez. So, Cross and Sudkamp they proposed this function as the modification function ok. Let us look at visually how do they look like. For the moment let us take B dash of x to be min of s comma B of x. Let us take this as the modification function.

So, what does it do? It takes the similarity value s and then thresholds it over B of y. So, in the formula it is x, but it does not matter. So, we are going to use this s to threshold B of y and get a new B dash. So, thresholding means essentially at this value s we are cutting it off. So, this is how the modified fuzzy set B dash will look like, if you use the operation given here B dash is min of s comma B. Let us use the modification function proposed by Cross and Sudkamp. It is given as min of 1 comma B x by s.

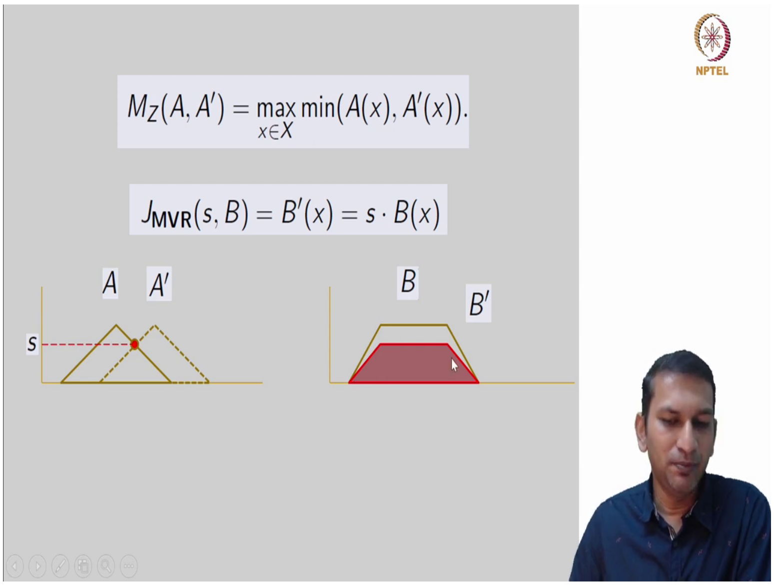
(Refer Slide Time: 18:53)



Perhaps, we will start with Morsi and Fahmy. So, the function that they have proposed is B dash is s dot B of x. So, immediately you see that this modification function J is nothing but that of a product. It is a product you know which is again a fuzzy logic connective. So, now how will it look like visually? So, we are having the same similarity value s and using this function we want to modify ok output, the consequent B.

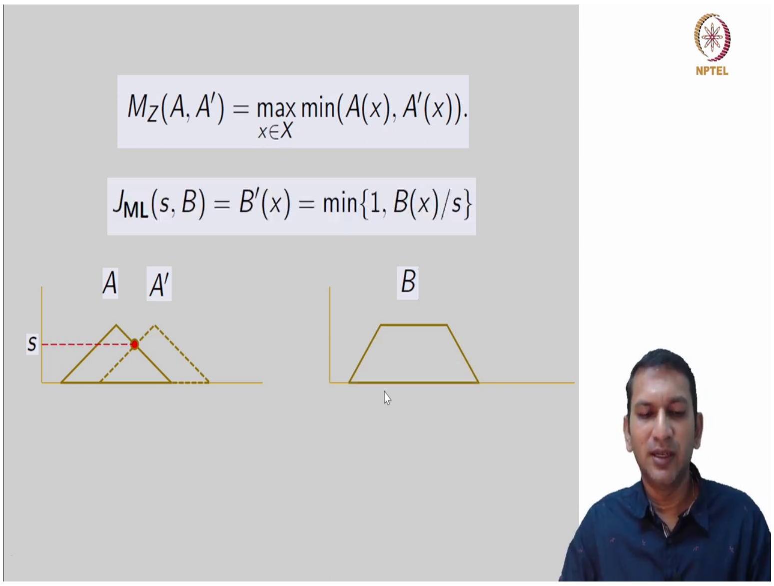
Note, that s here is a similarity value typically between 0 and 1. And, product operation essentially scales the fuzzy set P. Now, you will see here that the kernel of B is between these two points so, over this interval. So, at the point where B of x is 1, B dash essentially takes the value s that is the maximum value that B dash can take.

(Refer Slide Time: 19:51)



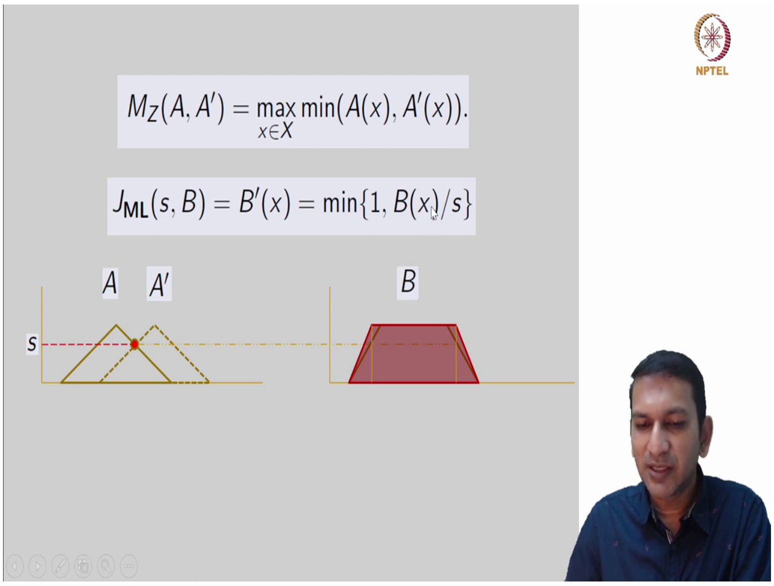
So, if you are using this modification function, you see that the support of B dash will be exactly the same of support of B. And, also B dash will be contained in B with respect to the point wise ordering.

(Refer Slide Time: 20:05)



Let us look at the modification function proposed by Cross and Sudkamp. So, it is given like this B dash is min of 1 by B by s. Now, once again it must be immediately clear to you this is the fuzzy logic connective. And what is it? It is the Goguen implication, minimum of 1 comma y by x. So, instead of y we are putting B of x here. So, this is again essentially an implication this is the which is a fuzzy logic connective. Now, when we use this modification function how would B dash look like? So, this is the similarity value s and we are going to modify it based on this.

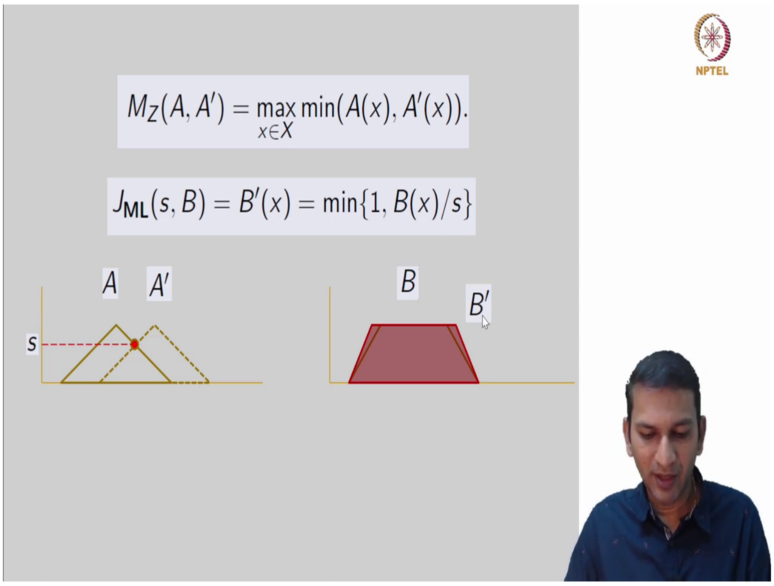
(Refer Slide Time: 20:48)



Now, look at this. We are looking at minimum of 1 comma B of x by s. You see here, at this point B of y is equal to s B of y is equal to s. Now, the moment it is s or above this on this entire interval, the membership value of every point in this interval is greater than that of s. So, essentially it is going to go above 1 and this operation acts as a threshold. So, it cuts it off at 1.

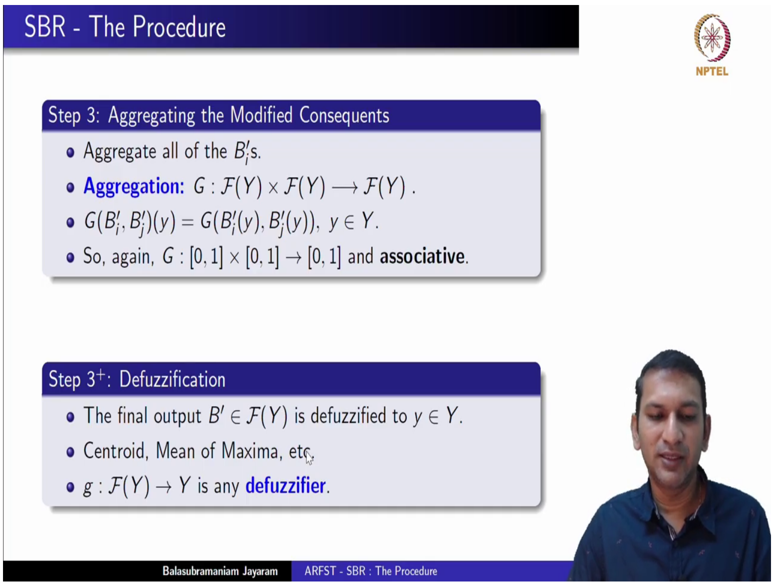
So, you will see that in the case that we are using the Goguen implication as the modification function which we call it as J ML here, the modification function proposed by Cross and Sudkamp. We find that the support of B perhaps does not change; however, B dash now contains B.

(Refer Slide Time: 21:35)



So, depending on the modification function that you use, the modified consequent B dash can either be contained in B or bigger than B with respect to the point wise ordering. Well, this is the second step. So, first step was to find to what extent the given A dash is similar to each of the antecedents and using the similarity value, we are going to modify the corresponding consequence B i to B i dash.

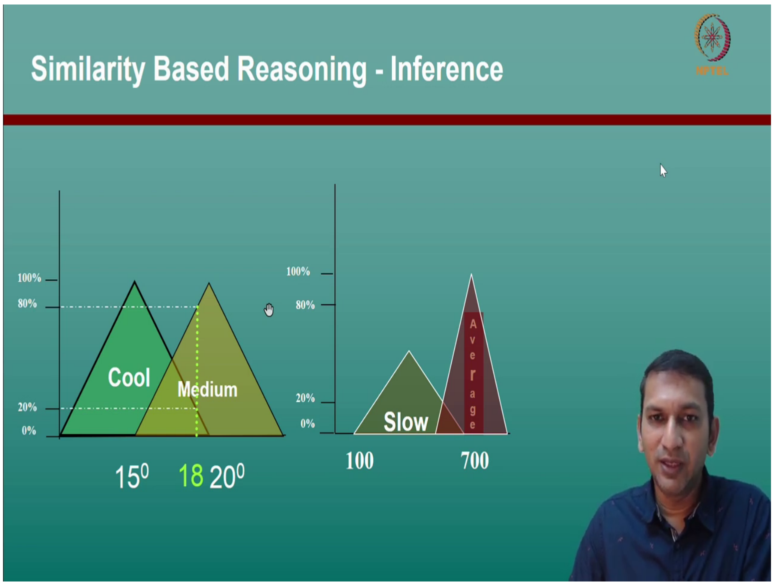
(Refer Slide Time: 22:07)



The third step is aggregating all these modified consequence. Aggregate all of these B i's which means we need an aggregation function. Once again, we denote it as G. It essentially takes two fuzzy sets from over Y and then gives you a fuzzy set on Y. Now, note that when you look into this, essentially once again G is also going to act only on the membership values of B i and B j, essentially it means you could still consider G to be a binary function of [0,1] to [0,1] which means again a fuzzy logic connective.

However, note that as in the case of FITA, you might have more rules than just two which means we would like this G to be associative. So, that you could aggregate them and you can do it order independently.

(Refer Slide Time: 23:01)

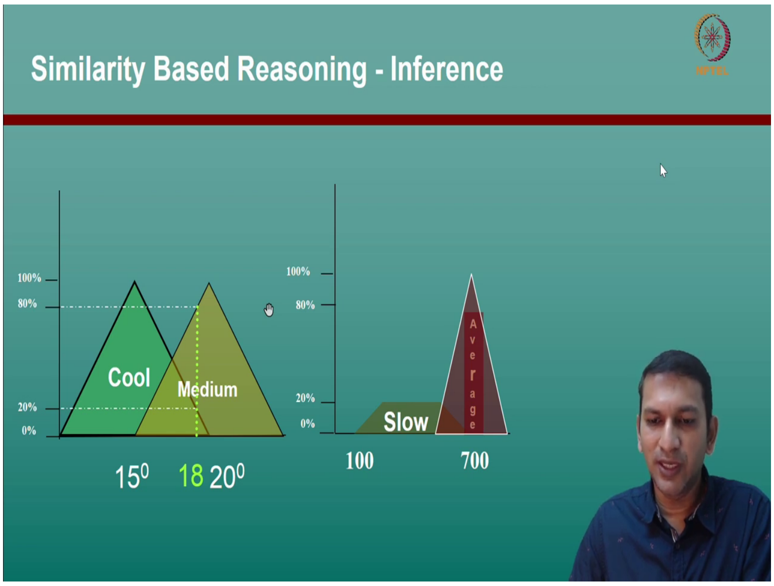


Now, let us revisit the example that we saw in the previous lecture. Now, the input given to us was 18 and we found 18 falls in the support of these two antecedents of the rules which state that if X is Cool, then Y is Slow. If X is Medium, then Y is Average. So, 18 degrees temperature falls in the support of these two antecedent fuzzy sets Cool and Medium; that how did we do the inference?

We looked at to what extent 18 degrees belong to both Medium and Cool. We found that it belong to the fuzzy set Cool, the antecedent Cool to degree 0.2 and that of Medium fuzzy set to degree 0.8. Clearly, we have considered the Ruspini partition. So, we see that the membership values add up to 1. This was the first step, we matched and found that we took the membership value itself as a similarity value, is it valid, is this assumption, is this proposition valid? We will see that yes, it is presently in few moments.

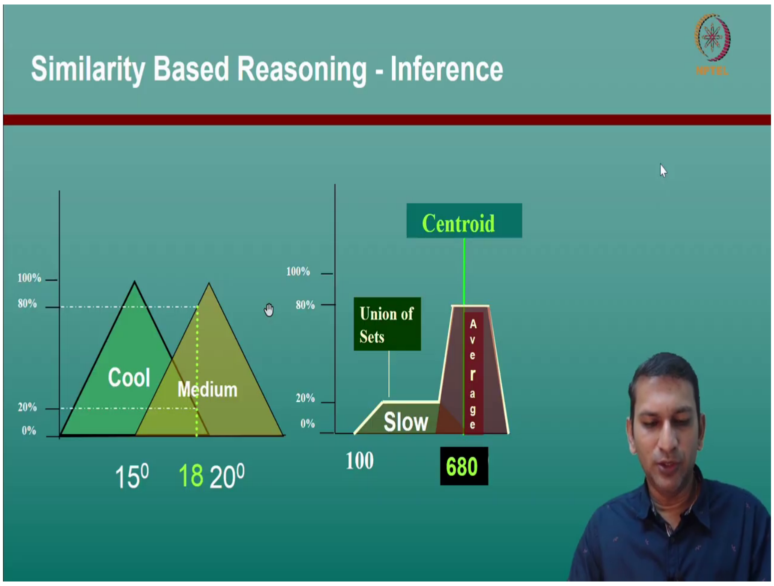
Now, taking these two similarity values 0.2 and 0.8, the next thing is to apply the modification function. And, if you look at it what we have done earlier is we have thresholded this consequence slow like this.

(Refer Slide Time: 24:30)



And, thresholded the Average, the consequent Average this way.

(Refer Slide Time: 24:37)



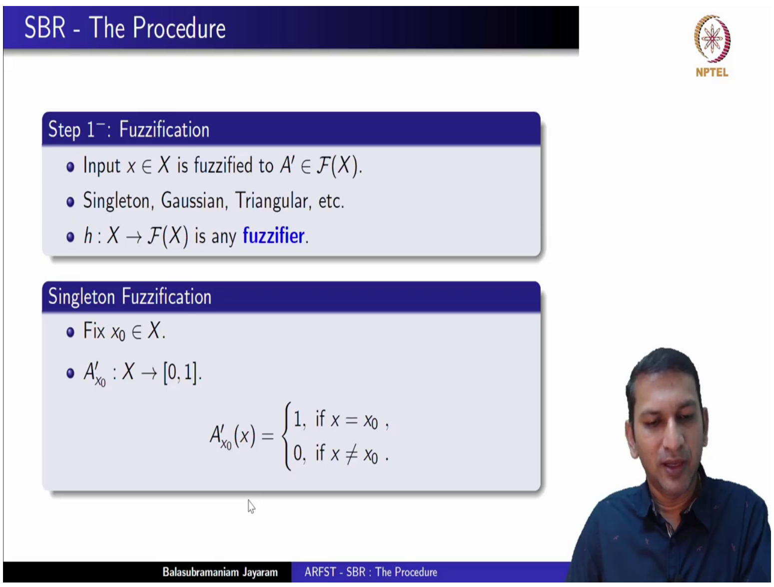
So, it is clear visually what we are doing is applying min, the min t norm as the modification function. So, that is the second step we have modified it. The third step involved taking the union of these two sets, essentially we applied the max operation which is the aggregation operation here. So, that is the third step, we have found out the aggregated B A dashes, the modified consequence of Slow and Average and we have aggregated them.

Now however, for the 18 degrees temperature that the control system has sense, we want to set the motor speed to a particular rpm and this fuzzy set is not going to help us, we need a number. So, we came up with this operation of applying centroid to come up with a value in Y. So, essentially what we have done is converted this fuzzy set on Y to a value on Y and this is the operation called defuzzification.

So, this is the next step that you need to apply, if you want to go back to one of the elements in y. What is defuzzification? Essentially, taking the final output B dash which is a fuzzy set on Y and defuzzifying into a value in Y. Centroid is one particular defuzzification. There are many more which we will see during the next few lectures this week.

So, you could look at it as a function small g which takes fuzzy set on Y and gives you a value on Y. But, there is one more thing to note, if you look at it what we have given is 18 degree centigrade. The temperature which is a real number, it is not a fuzzy set. However, we are applying fuzzy inference mechanism. So, is there something more that is happening here?

(Refer Slide Time: 26:36)

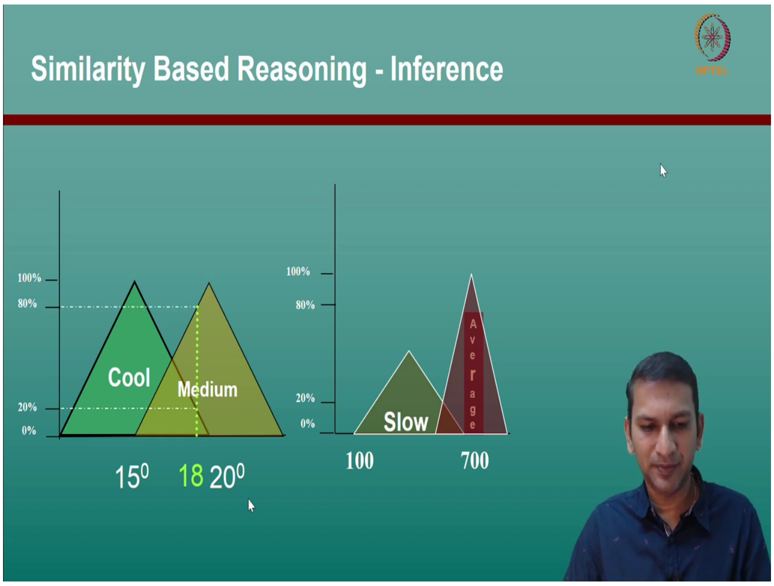


Yes, typically and often it is required that we have another preprocessing step, that is what was shown in the schemata as a fuzzifier. We often need to fuzzify a given value, if it is not presented as a fuzzy set itself. For instance, if you are given an x an element of X, we typically fuzzify it to a fuzzy set on X.

For this, we would use many types of fuzzification procedure, a singleton fuzzifier or Gaussian fuzzifier or triangular fuzzifier. So, essentially these are these this can be this step can be seen as a function h which takes a value on X and gives you a fuzzy set on X.

This we call the fuzzifier. Well, what is this singleton fuzzification? Let us pick an x naught from X, let us say this is the input that we want to give to the system and find the corresponding matching output. What we do is construct a fuzzy set A dash with respect to x naught as follows. This fuzzy set attains the value 1 at x is equal to x naught and everywhere else it is 0. So, essentially it is like a Dirac delta function, the characteristic function of the singleton set x naught.

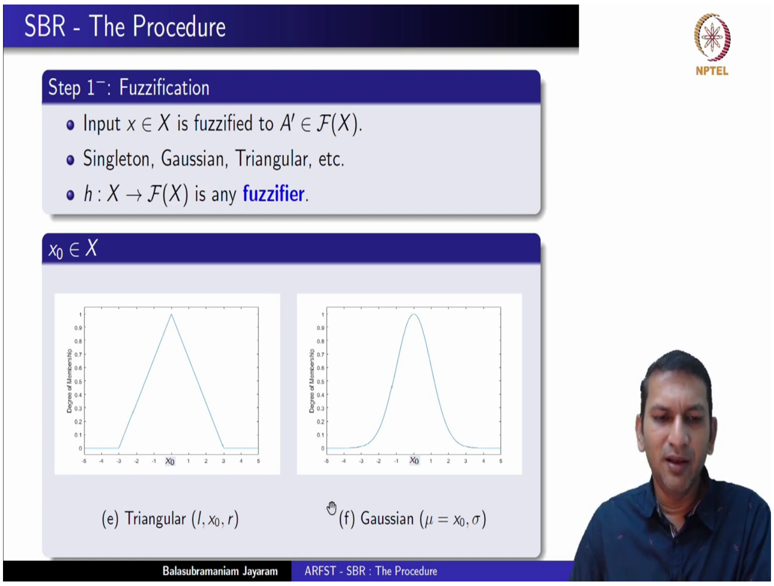
(Refer Slide Time: 27:51)



Now, let us look at this visual illustration once more. What did we give as the input? 18 degrees. And now what did we find? We are actually finding the membership degrees of 18 to the fuzzy sets Cool and Medium. Now, when you look at it like this, this is essentially the singleton fuzzy set that we have obtained by obtaining by applying the singleton fuzzifier to the point 18.

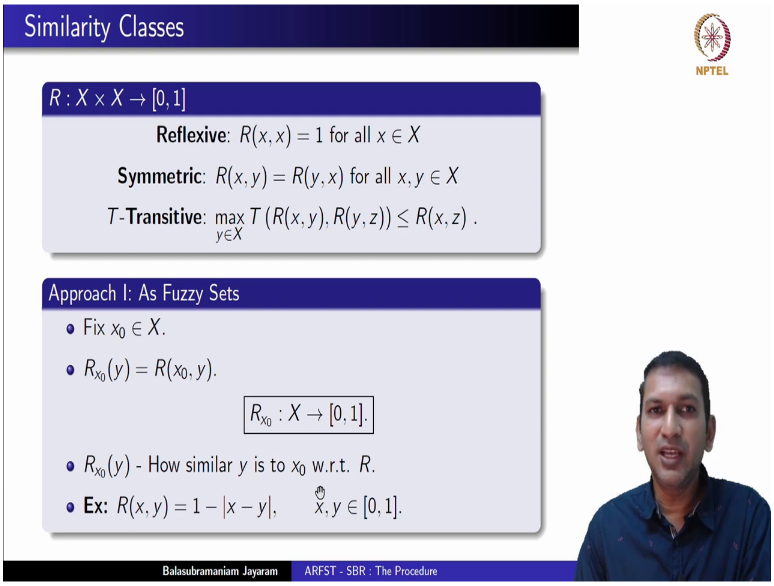
And, now you see that when you are using any one of those matching functions, take for example, the Zadeh's matching function, we will see that at this point at 18 if you consider it to be x naught, then only at x naught is equal to x is equal to x naught, you are going to get a spike a value 1. So, essentially applying the matching function to the singleton fuzzified fuzzy set and any of these antecedents is going to give you just the membership value of the corresponding antecedent fuzzy set. And, that is how we have found that it is 0.2 and this is 0.8. Well, what are the other kinds of fuzzification?

(Refer Slide Time: 29:05)



Well, we could have triangular fuzzification or Gaussian fuzzification. So, in triangular fuzzification clearly, we put the center point to be x naught and allow the left and right sides to taper, depending on how we want the fuzzification to be. Similarly, in the case of using Gaussian fuzzifier, the mu becomes x naught and we adjust the width by appropriately using the sigma value. But, perhaps there is one more thing that we could see here.

(Refer Slide Time: 29:38)

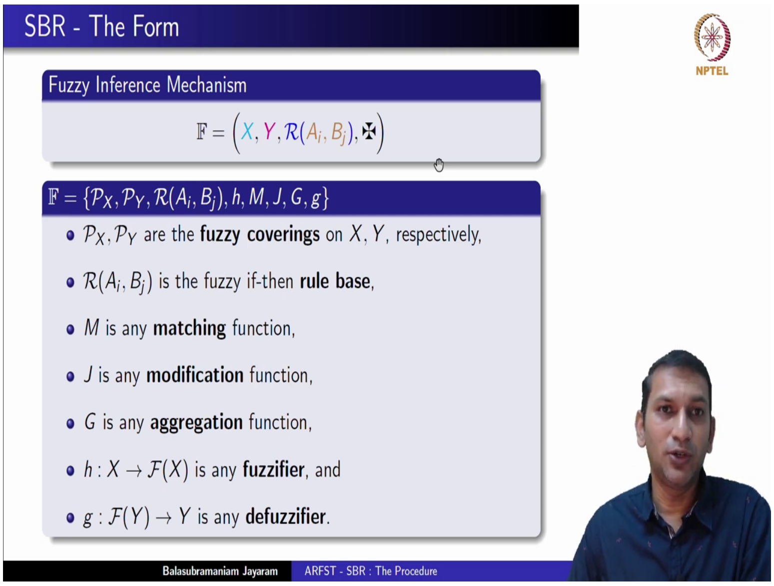


If you recall, we had discussed similarity relations. These are binary relations on x, fuzzy relations on x which are reflexive, symmetric and T-transitive. Later on, we also call them as fuzzy equivalence relations or T equivalence relation. At that point of time, we say that each row in the matrix can be looked at as a fuzzy set; that means, we fix an x naught and look at the a particular row in the similarity matrix, the relational matrix.

We know that it gives us a fuzzy set and we interpreted it like this R x naught of y is giving us a similarity value, how similar y is to x naught with respect to the relation R. So, you could look at fuzzification itself as a process where you are building not just a fuzzy set from a point, but with respect to some relation that you have in mind in the context with respect to the domain and with respect to the problem, that you are handling.

For instance, you might recall this was one particular singularity relation that we have used. If you put x naught here, considering this as to this to be the domain then essentially the fuzzification what you would get is the triangular fuzzification. So, every fuzzy fuzzifier essentially has some fuzzy relation behind it and it accordingly fuzzifies the point to a fuzzy set and it is not arbitrary.

(Refer Slide Time: 31:16)



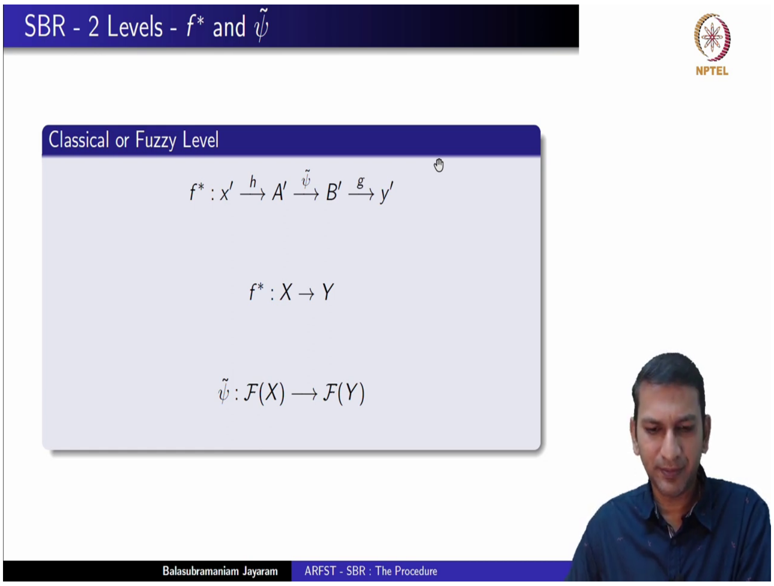
Well, we have seen that the general form of a fuzzy inference mechanism was given like this as a quadruple. It has this input and output fuzzy sets, the rule base and the inference engine itself, the operation that make up the inference. In the case of an SBR, the similarity based rules reasoning inference mechanism, we see that it has these many components.

P X and P Y are the fuzzy coverings on X and Y respectively. We have seen for a complete rule base. It is sufficient to have a fuzzy covering on X, but typically we also have fuzzy coverings on Y. R of A i, B j is the fuzzy if then rule base, where A is the antecedents coming from P X, the fuzzy covering on X. B j's are the consequence once again coming from P Y.

Now, we have restricted them to come from P X instead of just F X, for the reasons that we have seen before because we would like to have a complete rule base. Typically, in applications we would like to have a complete rule base. M which is used in the first step is any matching function F(X) cross F(X) to [0,1]. J is a modification function which again is a binary fuzzy logic connective, if you would; if you would like to choose it as such.

G is any aggregation function and h is the fuzzifier which takes an element of X and constructs a fuzzy set on X. g is the defuzzifier, does the opposite job, takes a fuzzy set on Y, a fuzzy set over a domain and maps it to some element in the domain.

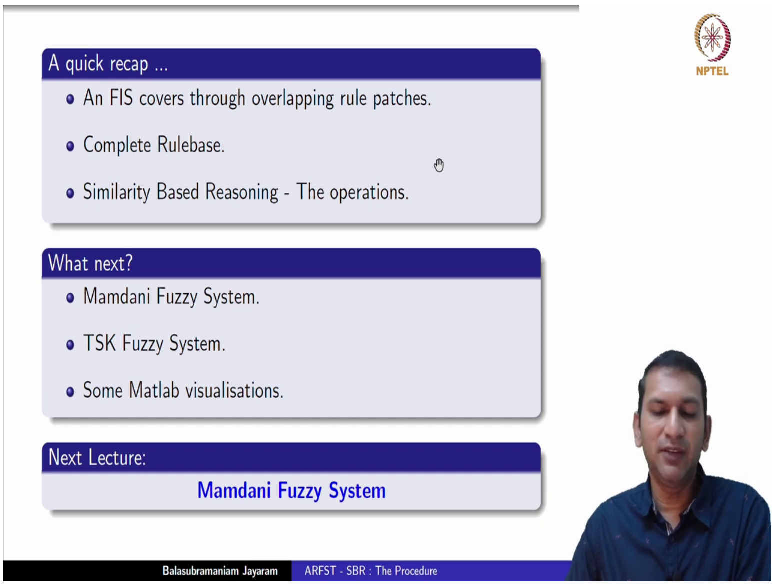
(Refer Slide Time: 32:54)



Now, we could look at fuzzy inference mechanisms themselves in at two levels, specifically the similarity based reasoning. Either at the classical level; that means, you are given an x dash which is coming from next, you apply the fuzzifier, get an A dash which is a fuzzy set on x. So, it belongs to F(X), apply psi tilde which we have seen as the fuzzy inference mechanism a map which maps F(X) to F(Y).

Get a B dash which is a fuzzy set on Y, apply the defuzzifier g and obtain a y dash. So, essentially f star is mapping from X to Y which is more like a classical function or you could also look at fuzzy inference mechanism, as just a mapping between fuzzy sets from the fuzzy sets on X to fuzzy sets on Y.

(Refer Slide Time: 33:44)

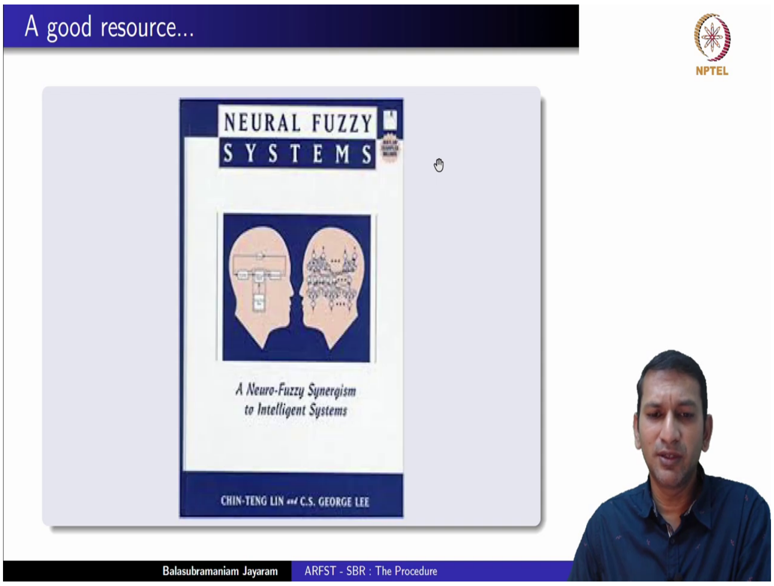


Well, a quick recap of what we have seen today. The most important point to note is that the fuzzy inference system covers the function that is trying to approximate through overlapping rule patches. This meant we often end up considering complete rule bases and we have seen the operations, the different steps involved in similarity based reasoning. What next? We have seen that there are two important major fuzzy inference systems that can be seen as similarity based reasoning fuzzy inference systems, that of Mamdani.

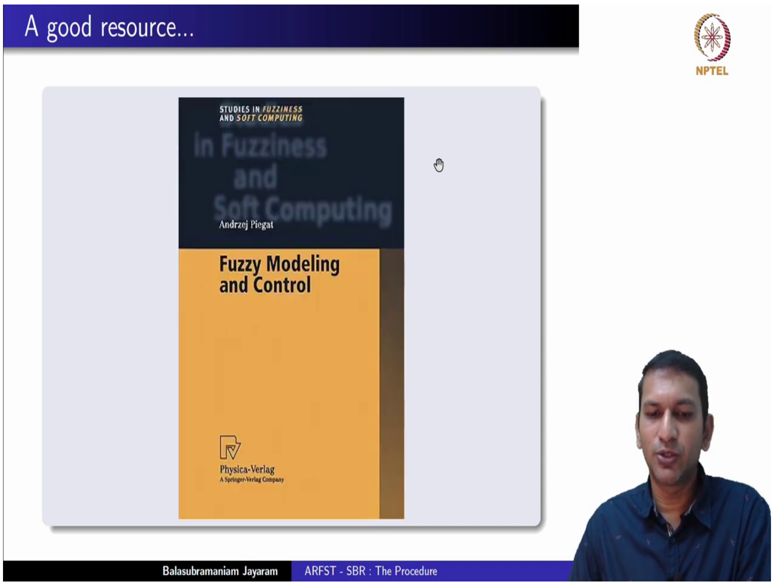
The one proposed by Ebrahim Mamdani and the Takagi Sugeno Kang fuzzy system. We will look into these two and also, we will take the aid of MATLAB, especially the fuzzy logic toolbox in MATLAB to see how to build the rule base, how to build the fuzzy inference system which can approximate any function that we are considering. Well, in the next lecture, we will specifically look at Mamdani fuzzy systems.

(Refer Slide Time: 34:48)

(Refer Slide Time: 34:54)



(Refer Slide Time: 34:57)



Once, again a good resource for the topics that we have covered in this lecture are the books of Passino and Yurkovich, that, of C T Lin and George Lee and, also that of Professor Piaget. Glad that you could join us in this lecture. Hope to see you soon in the next lecture.

Thank you again.